THE BEST SELLING COMPUTER PROJECTS MAGAZINE SEPTEMBER 1984

ELECTRONICS & COMPUTING

59 Robot 1

91 PRINTER INFO

OVER £700 OF PRIZES TO BE WON INSIDE

VIEW PRINTER DRIVER-OFFER

AN EMAP PUBLICATION

USA $2.95 Germany DM.9.90 90p

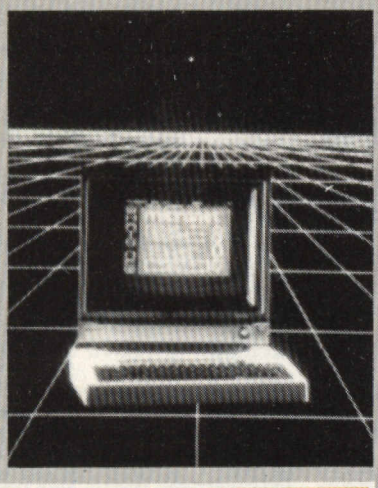
MICRO COMMUNICATIONS
Logging on to the future

HOW MANY HERTZ? · BBC & SPECTRUM FREQUENCY METERS

EXPLORING THE HIDDEN POWER OF LOGO AND FLEX

ALL ABOUT ALLOPHONES · PRINTER SURVEY SPECIAL

ELECTRONICS & COMPUTING
Contents
Vol. 4 Issue 9



COVER PHOTO BY ROB BRIMSON

Electronics & Computing Monthly
Scriptor Court, 155 Farringdon Road,
London, EC1R 3AD

Editorial 01-833-0846
Editor Gary Evans
Assistant Editor William Owen
Production Editor Liz Gregory
Administration Serena Hadley

Advertising 01-833-0531
Advertisement Manager Anthony Herman
Chief Executives Richard Jansz
Francis Lee
Classified Tracey Keighley
Assistant Yvonne Moyser

Production 01-833-0531
Art Editor Jeremy Webb
Make-up Time Graphics
Publisher Alfred Rolington

Distribution
EMAP National Publications

Published by
EMAP Business and Computer Publications

Printed by
Riverside Press, England

Subscriptions
Electronics & Computing Monthly, (Subscription Department), Competition House, Farndon Road, Market Harborough, Leicestershire.

Electronics & Computing Monthly is normally published on the 13th day of each month.

PROJECTS

FEATURES

REVIEWS

PLUS

And within *Your Robot*

News of a new low-cost robotic development kit, a closer look at how Prism's Movits perform plus a round up of all the latest news from this new and fascinating field of computing.

We hope you'll take the time to complete our reader survey that appears on pages 74 and 75. We know this can be a bit of a chore but surveys such as this are a great help to us and ensure that E&CM meets the needs of our readers. As an added incentive we're giving away £500 to one lucky person who's completed entry is drawn from the traditional hat later in the year.

See page 58 for details of this month's competition with a first prize worth over £200.

NEWS NEWS NEWS NEWS

First multi-tasking Electron FORTH

Skywave Software in the form of David Husband and colleagues, has just launched Multi-Forth for both the BBC and Electron computers. We shall be taking a close look at the package in the next issue of *E&CM* but here are a few brief details of this Forth implementation.

Multi-Forth 83 has been specially written and is not just a re-hash of existing Forth-79 code. it conforms to the recently released Forth-83 standard and is one of the most sophisticated Forth packages on the market.

One of the major features of the Skywave Forth is that it supports multi tasking. It allows a number of Forth programs to run simultaneously and transparently of each other. Each task is placed in a queue and the maximum number of tasks in the queue is no less than 28.

For anyone exploring robotics systems or experimenting with the concepts of computer control, Multi-Forth offers facilities that are bound to be of immediate application.

For more details of either the BBC or Electron version of the package contact Skywave Software at 71 Curson Road, Boscombe, Bournemouth, BH1 4PW. Telephone: 0202-302385.

In search of entertainment

As micros find their way into an increasing number of homes it is not surprising that the mass media are paying more attention to computers in both home and business applications. The BBC led the way with the 'Making the most of the Micro' series and have recently followed this with a series on computers in control applications and one on business applications. In addition the Beeb have been putting out the 'Chip Shop' on Radio 4, although some would argue as to whether or not this could be classed as mass media.

The commercial TV stations have also started to pay attention to the home micro scene. Yorkshire TV produced 'Me and my micro' which recently finalised its run and Thames TV have two series of 'Database' behind them. The approach adopted by the production teams behind these programmes is interesting. In general a programme that goes out on one of the major networks must be 'entertaining' if it is to pull in the sort of audiences that are expected. This is, it can be argued, of greater important in the case of commercial TV who need large audiences to satisfy their advertisers.

The quest for entertaining programming leads to a considerable emphasis on making the material visually interesting, which is all well and good, but sometimes leads to other less desirable features. The most irritating of these is when concepts are simplified to such an extent that they cease to make any technical sense. This simplification assumes that the viewer is incapable of assimilating any information that might stretch the brain to any extent. In one programme trying to explain the methods by which computers communicate over 'phone lines, a modem was described as a filter that overcomes noisey lines. This is taking the simplification process too far and indeed is a gross inaccuracy.

The BBC's Open University programmes have, for a number of years now demonstrated that it is possible to produce programmes that are both interesting, and in many cases entertaining and accurate. They have shown that, in dealing with complex subjects it is possible to produce material that meets the requirements of mass media broadcasting while keeping within the bounds of technical accuracy.

Channel 4 is said to be planning an 'up-market' computer programme this autumn and it can only be hoped that this series does not follow the route of some of TV's recent output.

GARY EVANS

S-22 – a Dragon tamer

The Ashby computer centre have recently begun to distribute a machine code programming aid for the Dragon that goes under the name of S-22. The software was not developed specifically for the consumer market but was written by people who were involved with the development of commercial 6809 systems in order to make their work easier. It was almost by accident that the computer centre spotted the toolkit and realised its potential as an aid to owners of Dragon computers.

S-22 comes as a plug in cartridge and is supplied with extensive documentation that takes the form of two manuals. One is a guide to the use of S-22 and the other is Appendix C which gives a lot of useful information about the operation of the Dragon, essential if full use is to be made of the computer's facilities.

S-22 provides extensive trace and debugging aids and allows machine code routines to be quickly up and running. The package should be of particular use to those who wish to incorporate small machine code sections within their BASIC programs as it provides the means to accomplish this more readily than traditional assembler packages.

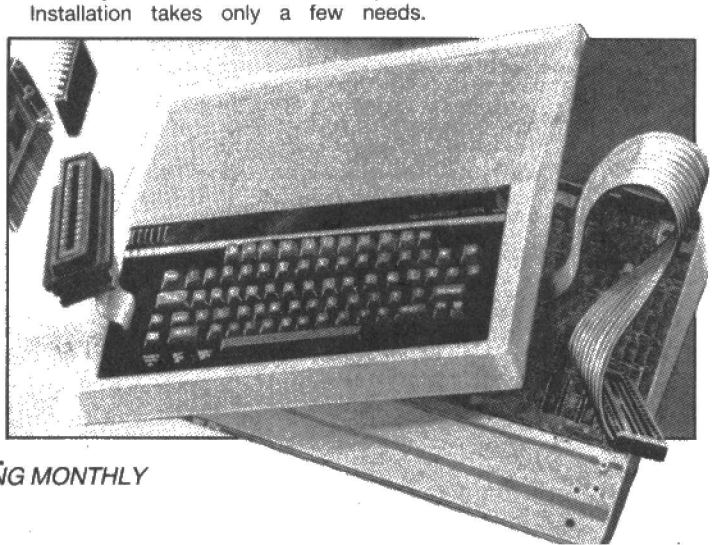
S-22 is available direct from the Ashby Computer Centre at 186 High Street, Ashby, Scunthorpe.

Easing ROM replacement

Like so many good ideas, Acorn's concept of the sideways ROM sockets on the BBC micro seems so obvious in retrospect. The company though cannot have foreseen the way in which the idea has taken off over the past couple of years with the result that today there is such a vast range of sideways ROM software available for the computer. Many users will have in excess of the four ROM based packages that can be accommodated in the BBC micro and have in the past had to resort to a ROM expansion board if they wanted all their software resident simultaneously.

A new product from Viglen provides an alternative. It marries the sideways ROM concept with that of cartridge based software and in effect allows users to make their own cartridges. The basic pack provides a ROM carrier together with a cartridge slot that fits into the infamous 'hole' to the left of the beeb's keyboard.

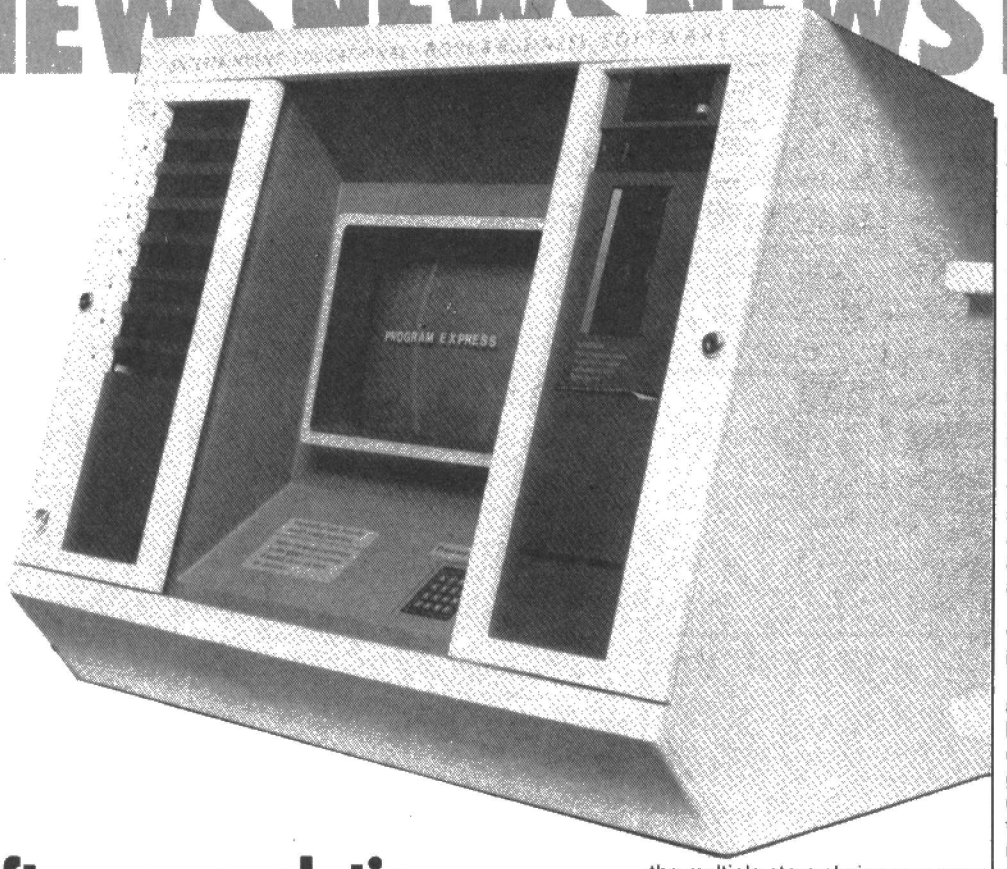
Installation takes only a few minutes and a great deal of attention to detail is evident in the design of the system. Anyone with a growing collection of ROM software and a reluctance to continually take the 'lid' off their computer to install new software could find the Viglen system an ideal solution to their needs.



Wana buy a Spectrum

It is reported that Sinclair are scheduled to produce in the region of 600,000 Spectrums in the period from now until Christmas. This two year old design will face an increasingly crowded market and despite the wealth of software available for the machine (an attractive selling point for any micro) Sinclair may find many first time buyers who until now have turned to the Spectrum, opting for some of the computers that have recently come onto the market.

Sinclair's main weapon in any sales battle will be a price cutting exercise, in fact some price reduction would seem to be an odds-on certainty in the not too distant future.



Software revolution

If Program Express are to be believed, the way in which software is distributed in the UK could radically change over the next couple of years. From what *E&CM* has seen we see no reason to doubt the company's claim.

The present way of producing, distributing and retailing software has a number of in-built problems. Not the least of these is the need for all parties in the chain to tie up large sums of money in maintaining an adequate stock level. For the publishers of software, the need to commit money in this way has contributed to the downfall of a number of companies in the past – the problem often surfaces when a company is faced with a large level of stock returns. For the retailer the need to minimise capital outlay inevitably means that the range of stock carried in any outlet is restricted to a comparatively narrow range of items. Even the largest branches of the multiple store chains may carry only 300 – 500 titles from the many thousands of those available.

The solution offered by Program Express, in conjunction with the ITS company of America is the software machine. This is an EDOS (Electronic Distribution Of Software) system and while this concept is not new Program Express have refined the system to make it an attractive alternative to those involved with software distribution.

For an outlay of about £95 per week a retailer is supplied with a package that consists of a monitor screen, hard disc unit with processor and a tape duplication facility. The unit's hard disc will hold 1000 different programs assuming an average 30K length.

A customer is able, via a menu driven sequence of instructions, to select first of all those titles available for his computer and then to the subject area in which they are interested. When a choice of package is made it is possible to read a review of the item, to view a still frame from the software (important in the case of games) or to purchase the item. If the purchase option is selected, the machine will summon an assistant who will enter an authorisation code before inserting a tape into the duplicator.

The duplicator copies the software at 10 times normal speed and incorporates a verification process.

The system removes the need for any link in the distribution chain to hold high levels of stock and has other advantages. The problem of getting new titles into the shops as quickly as possible is solved by virtue of the fact that each software machine can go on line to a central computer to download any item of software. The system also means that local outlets can begin to carry items of special or minority interest without worrying about tying up cash in slow moving lines. It should also mean that high cost business packages will be readily available in local outlets.

Most of the major retail chains have expressed interest in the system nd, as the majority of home micro software is sold through the stores, if they decide to adopt the system then a distribution revolution could be just around the corner.

Versatile plotter

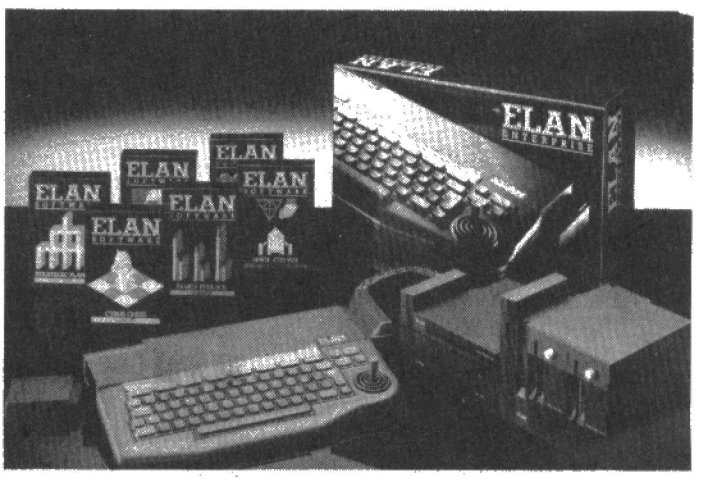
Parfitt Electronics have introduced a new plotter aimed specifically at the BBC market. The peripheral can, however, be used for much more than just tracing or creating specific pictures. Besides the normal accommodation for holding pens the carriage is also designed to carry other attachments like an opto-sensor which is used for scanning drawings and a drill attachment which may be used for inscribing upon softer materials like polystyrene, foil and even for the boring of bolsa wood. The company hope that these additional facilities will particularly interest education authorities with special emphasis upon the blind as shapes that can be felt may be created. The plotter costs around £270.00 but the full kit complete with all attachments, power supply and software comes to £490.00. Parfitt Electronics are at 6 View Road, London N6 and they are on 01-348 1973.

Elan/Flan enterprise – here at last!

Those of you with long memories may well remember reports of the press launch of the Elan computer that appeared in the autumn of last year. At that time the computer was promised for April of this year. In this industry it came as no surprise when the machine failed to appear on time and indeed after the non-arrival of the machine some people were beginning to write the Elan off.

These doubters are about to be proved wrong. The computer, now called the Enterprise is due for launch in early September and it is rumoured that the company behind the machine plan to have 100,000 in the field by this Christmas. The computer is to retail at just under £200 and aims to offer a natural upgrade for Spectrum owners as well as offering an attractive package for first time buyers although the Amstrad CPC464 will provide tough competition in this area.

The Enterprise has a number of attractive features not the least of which is the provision of a wordprocessor package as part of the machines firmware (a la Commodore Plus4). The designers have placed the emphasis on quality and the video and sound capabilities of the machine put many of the current crop of micros to shame. Our mole also had a few more comments on the potential of the machine but to go into these at this stage might make it possible to trace our informant, so for now thats it. Keep reading *E&CM* for further details of the Enterprise though.



HOW MANY HERTZ

Paul Beverley shows that, with a few lines of cunning software, the BBC micro can be made to function as an auto ranging frequency meter. Adding some extra hardware will extend the versatility of the design.

In the last couple of articles we have looked in some detail at the way the counter-timers on the 6522 versatile interface adaptor (VIA) work. This month we will show how to put those ideas into practice in order to make the BBC microcomputer act as an auto-ranging frequency meter. If the pulses arriving at the computer are TTL pulses ie 0 to +5V, then all you need to do is enter the program and you will have a frequency meter which can measure pulses with a frequency between 0.5 Hz and 480 kHz.

The resolution of the measurements is in the region of 1 part in 30,000, but the absolute accuracy depends, of course, on the accuracy of the crystal clock in the computer. The clock frequency, as mentioned last month, varies by roughly 1 part in 5,000 as the computer warms up. However, once it reaches a steady temperature, it is stable to within about 1 part in 100,000 which is quite adequate when compared with the resolution. Correction does have to be made for the fact that, although it drifts very little, the actual value of the clock frequency may be in error by as much as 0.2%. Therefore, if the computer's clock frequency can be measured by a commercial frequency meter, this corrected value can be included in the program at line 720. The techniques used for measuring the frequency are firstly to count the number of pulses coming in during one centisecond (lines 110, 120) and then, depending on the value returned, select one of three possible measuring techniques (130 – 150).

PROCvlf

This routine is active when the frequency is below 38 Hz. The BASIC part occupies lines 390 – 510, and the machine code routines are at lines 2090 – 2710. This uses the technique, described last month, of taking a rough measurement of the time period using the centisecond interval timer and then taking a measurement on Timer 1 in the free-running mode with the interrupts disabled. You need not worry about how many times the timer "goes round" since this is ascertained from the rough timing which is done first.

Even without using these routines the program is still capable of working down to 8 Hz and so, if you do not want to go lower than that, you can delete lines 140, 390 – 510 and 2090 – 2710.

PROClf

This is used for the major part of the frequency range up to 100 kHz. The BASIC section is lines 240 – 390, and the machine code, 1080 – 1570. The technique used is to take a rough measurement first by measuring the time of one pulse, calculating the number of pulses that should arrive in a time of something less than 130 milliseconds (&1FFFF microseconds), and then use Timer 1 to measure the actual length of time in microseconds for that number of pulses to arrive, as counted in by Timer 2.

PROChf

For the highest frequencies (100 kHz +) we count the number of pulses arriving in a specific period of time. The program first of all works out how many centiseconds to count for in order to get a count of up to &1FFFF (lines 180 – 220). The pulse counting routine (lines 1590 – 2070) is the same as is used for the initial sample to select the most appropriate measurement technique. The idea is to wait for a time-out on Timer 1 on the internal VIA (as used by the operating system for the centisecond clock facilities), start Timer 2 counting the in-coming pulses, then wait for and count the Timer 1 time-outs and then read the number of pulses as registered by the Timer 2 registers.

The upper limit to the speed at which this can work is set by the fact that these pulses are really supposed to be synchronised to the 1 MHz clock. The in-coming pulses are independent of the computer's clock, and so as they approach the theoretical maximum clocking speed of 500 kHz, if the mark-space ratio of the pulses is not exactly one-to-one, then the VIA begins to miss pulses, and the value read decreases as the actual frequency increases. The practical limit is something like 480 kHz for a reasonably evenly spaced pulse.

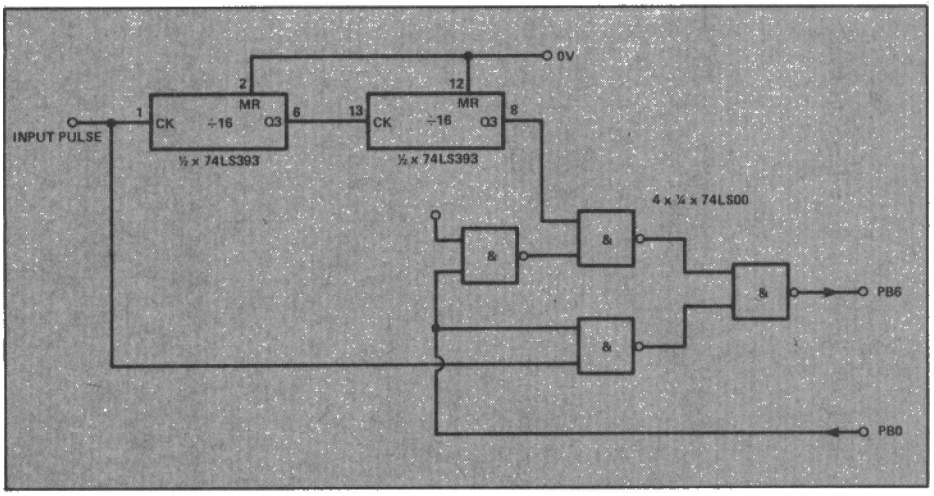


Figure 1. Using this prescaler circuit the range of the frequency counter can be extended to 45MHz.

Extending the Frequency Range

It is unlikely that anyone will want to measure, on an automatically ranging meter, frequencies of less than 0.5 Hz, but it would be useful to be able to extend the upper end of the frequency range. This can be done by adding a divide-by-16 or divide-by-256 stage before the input to PB6, and the direct signal or the divided down signal can be selected automatically by using say PB0, in response to the initial sample value. A suitable circuit is given in **Figure 1.**

Using just one divide-by-16 extends the range of 7 MHz, and using two would extend it to something like 125 MHz. The

problem here is that low power Schottky TTL gates (74LS series) only work up to about 45 MHz, but that is probably sufficient for most purposes.

There are two reasons for using binary rather than decade counters. First of all, since the frequency display value is calculated and not displayed directly, it makes no difference whether you have to divide the answer you get by 256 or 100. Secondly, as we have just said, it is better at the highest end of the frequency range to have a one-to-one mark space ratio, which the decade counter would not provide.

Signal conditioning

If the signal to be measured is not already at TTL voltage levels, then some signal conditioning is necessary. It is difficult to provide a suitable amplifier which will operate all the way from 0.5Hz to 45MHz, and so different circuits will be needed depending on the application. There are a number of designs published in magazines and books which you could use for the upper frequency end, and for the lower frequency a simple operational amplifier circuit would do.

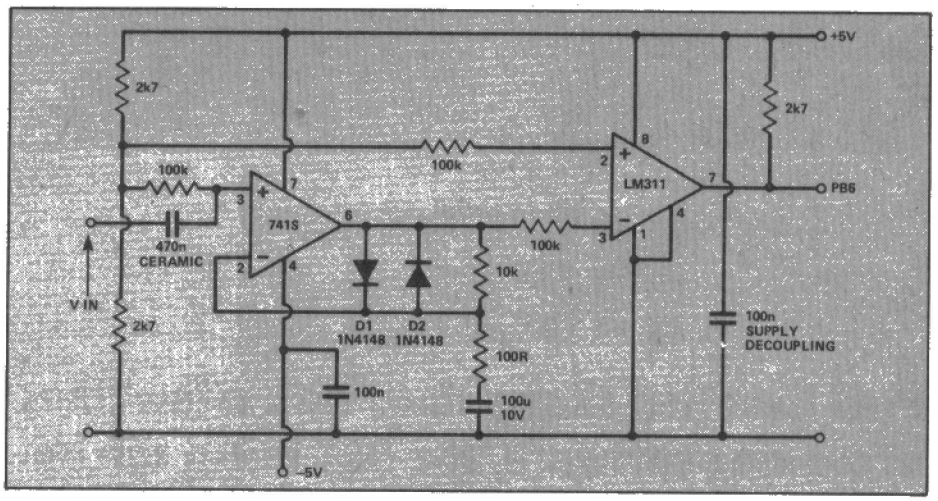


Figure 2. Circuit of a signal preconditioner suitable for medium frequency appreciation.

The circuit shown in **Figure 2** will work from about 8 Hz to 70 kHz with a sensitivity of 1 millivolt, to 150 kHz at 2 millivolts, 250 kHz at 5 millivolts and 480 kHz at 10 millivolts

Program 1 measures and displays the frequency of periodic pulses applied to the PB6 line on the User Port from less than 1 Hz to more than 400 kHz.

Next month, we turn our attention to the shift registers and use them to make an elementary logic state analyser costing under five pounds. ■

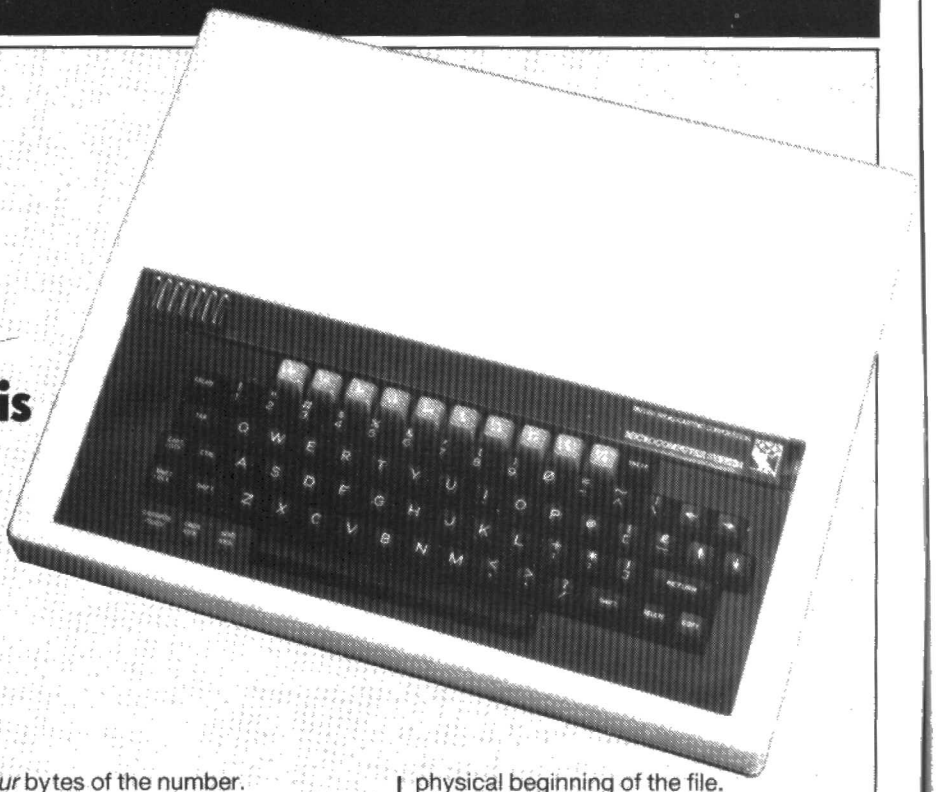
PROGRAM 1

```
 10 MODE5
 20 PROCinitialise
 30 REPEAT
 40   PROCmeasurefrequency
 50   IF ave THEN PROCaverage
 60   PROCdisplay
 70   UNTIL INKEY$(0)<>" "
 80 END
 90
100 DEF PROCmeasurefrequency
110 Y% = 1
120 CALL count_pulses
130 IF C%>K% PROChf:ENDPROC
140 IF C% = 0 PROCvlf:ENDPROC
150 PROClf
160 ENDPROC
170
180 DEF PROChf
190 Y% = targetcount%/C%
200 CALL count_pulses
210 frequency = C%*j/Y%
220 ENDPROC
230
240 DEF PROClf
250 pulses% = 1
260 PROCtimepulses
270 IF C%>&FFF0 ENDPROC
280 pulses% = targetcount%/C%
290 PROCtimepulses
300 ENDPROC
310
320 DEF PROCtimepulses
330 X% = (pulses%-1) MOD 256
340 Y% = (pulses%-1) DIV 256
350 CALL time_pulses
360 frequency = pulses%/C%*k
370 ENDPROC
380
390 DEF PROCvlf
400 base% = &10000
410 A% = 0 : REM start interval timer at zero
420 CALL rough_time
430 R% = A%*10000 MOD base%
440 IF R%<T% OR R%>(base%-T%) base% = 50000
450 X% = (base% - 2) MOD 256
460 Y% = (base% - 2) DIV 256
470 CALL fine_time
480 ti% = A%*10000 DIV base% * base%
490 time% = ti% + base% - B%
500 frequency = k/time%
510 ENDPROC
520
530 DEF PROCaverage
540 IF INKEY$(5) = " " total = 0:num = 0
550 total = total + frequency
560 num = num + 1
570 frequency = total/num
580 ENDPROC
590
600 DEF PROCdisplay
610 value = frequency
620 IF frequency>K% THEN value = frequency/K%
630 PRINT TAB(4,10) LEFT$(STR$(value) + "      ",8);
640 IF frequency>K% THEN PRINT" kHz" ELSE PRINT" Hz "
650 ENDPROC
660
670 DEF PROCinitialise
680 VDU19;4;0;
690 INPUT"Averaged values",A$
700 IF ASC(A$) = 78 THEN ave = 0 ELSE ave = 1
710 CLS
720 fclock = 0.9988766 : REM value of the 1 MHz clock
730 targetcount% = &1FF00
740 K% = 1000
750 T% = 10*K%
760 j = 100*fclock
770 k = K%*K%*fclock
780 VDU23;10,32,0;0;0;
790 count% = &40C
800 C% = 0
810 X1 = 220:Y1 = 650:X2 = 1070:Y2 = 740
820 MOVEX1,Y1:DRAWX2,Y1:DRAWX2,Y2
830 DRAWX1,Y2:DRAWX1,Y1
840
850 PB = &FE60
860 T1L = &FE64
870 T1H = &FE65
880 T2L = &FE68
890 T2H = &FE69
900 ACR = &FE6B
910 ?ACR = &60 :REM T1 free-run, T2 count-down
920 VIA_B_flags = &FE6D
930 VIA_A_flags = &FE4D
940 T1 = &FE44
950
960 B% = 0
970 PB = &FE60
980 T1L = &FE64
990 T1H = &FE65
1000 ACR = &FE6B
1010 OSWORD = &FFF1
1020 parameter_block% = &404 : REM A%
```

```
1030
1040 FOR opt% = 0 TO 2 STEP2
1050 P% = &C00
1060 [OPT opt%
1070
1080 .time_pulses
1090 \***********
1100
1110 SEI
1120 LDA #0             \ Make 3rd byte
1130 STA count% + 2     \ of C% = 0.
1140
1150 LDA #&FF           \ &FF ready for timing
1160 STA T1L            \ on Timer 1.
1170 STX T2L            \ X% = low byte of pulses%
1180
1190 LDX #&FF           \ ready to put in T1high.
1200 LDA #&20           \ ready for BIT on T2flag.
1210
1220 BIT PB             \ Wait until PB6 is high.
1230 BVC P%-3
1240
1250 BIT PB             \ Wait until PB6 is low.
1260 BVS P%-3
1270
1280 STX T1H            \ Start Timer 1.
1290 STY T2H            \ Start to count pulses.
1300
1310 .wait
1320 BIT VIA_B_flags    \ Have enough pulses been
1330 BEQ wait           \ recorded by Timer 2?
1340
1350 LDX VIA_B_flags    \ To see if T1 has timed out.
1360 LDA T1L            \ Low byte of time.
1370 LDY T1H            \ High byte of time.
1380
1390 CMP #3             \ If T1L is less than 3,
1400 BCS over_2         \ then T1H has over-shot.
1410 INY
1420
1430 .over_2
1440 EOR #&FF           \ Subtract from &FF.
1450 STA count%
1460 TYA
1470 EOR #&FF           \ Now the high byte.
1480 STA count% + 1
1490 TXA
1500 AND #&40           \ Had T1 timed out?
1510 CMP #&40
1520 BNE cli
1530 INC count% + 2     \ If so add one to next byte.
1540
1550 .cli
1560 CLI
1570 RTS
1580
1590 .count_pulses
1600 \***********
1610
1620 SEI
1630 LDX #0             \ Put zero in the
1640 STX count% + 2     \ 3rd byte of C%.
1650 DEX
1660 STX T2L            \ Low byte of pulse count.
1670 LDA #&40           \ Ready to BIT for T1.
1680
1690 .ready
1700 BIT VIA_A_flags    \ Wait for start of
1710 BEQ ready          \ centisecond interval.
1720 STX T2H            \ Start to count with Timer-2.
1730
1740 .next
1750   LDX T1             \ Reset flag on internal
                          VIA Timer 1.
1760
1770 .stop
1780 BIT VIA_A_flags
1790 BEQ stop           \ Wait for end of centisecond.
1800 DEY
1810 BNE next       \ Wait for the Y'th centisecond.
1820
1830 LDX VIA_B_flags    \ Get flag register.
1840 LDA T2L            \ Low byte first,
1850 LDY T2H            \ then high byte.
1860
1870 CMP #3
1880 BCS more_than_two
1890 INY
1900
1910 .more_than_two
1920 EOR #&FF           \ Subtract from &FF.
1930 STA count%         \ Low byte
1940
1950 TYA
1960 EOR #&FF
1970 STA count% + 1     \ High byte
1980
1990 TXA
2000 AND #&20           \ Had it timed out?
2010 CMP #&20
2020 BNE finish
2030 INC count% + 2     \ If so increment next byte.
2040
2050 .finish
2060 CLI
2070 RTS
2080
2090 .rough_time
2100 \**********
2110
2120 BIT PB             \ If PB7 = 0, wait.
2130 BVC P%-3           \ i.e. until PB7 = 1.
2140
2150 BIT PB             \ If PB7 = 1, wait.
2160 BVS P%-3           \ i.e. until PB7 goes to 0.
2170
2180 .write
2190 LDX #parameter_block% MOD 256
2200 LDY #parameter_block% DIV 256
2210 LDA #4
2220 JSR OSWORD
2230
2240 BIT PB             \ If PB7 = 0, wait,
2250 BVC P%-3           \ i.e. until PB7 = 1.
2260
2270 BIT PB             \ If PB7 = 1, wait,
2280 BVS P%-3           \ i.e. until PB7 goes to 0.
2290
2300 .read
2310 LDX #parameter_block% MOD 256
2320 LDY #parameter_block% DIV 256
2330 LDA #3
2340 JSR OSWORD
2350
2360 RTS
2370
2380 .fine_time
2390 \*********
2400 SEI                \ Don't interrupt the timing!
2410
2420 STX T1L            \ Set timer 1 low latch.
2430
2440 BIT PB             \ If PB7 = 0, wait.
2450 BVC P%-3           \ i.e. until PB7 = 1.
2460
2470 BIT PB             \ If PB7 = 1, wait.
2480 BVS P%-3           \ i.e. until PB7 goes to 0.
2490
2500 STY T1H            \ Start Timer 1.
2510
2520 BIT PB             \ If PB7 = 0, wait.
2530 BVC P%-3           \ i.e. until PB7 = 1.
2540
2550 BIT PB             \ If PB7 = 1, wait.
2560 BVS P%-3           \ i.e. until PB7 goes to 0.
2570
2580 LDX T1L            \ Catch low byte first,
2590 LDY T1H            \ then high byte.
2600 CPX #3             \ Check if it's 2, 1 or 0.
2610
2620 BCS more_than_2
2630 INY                \ T1H must have decremented
2640                    \ AFTER T1L was read.
2650
2660 .more_than_2
2670 STX &408           \ REM = B%
2680 STY &409
2690
2700 CLI
2710 RTS
2720 ]
2730 NEXT
2740 ENDPROC
```

RANDOM ACCESS

Adam Denning of Micronet 800 continues his exploration of random access filing on the BBC micro.



Now that we've got to grips with using standard operating system calls to open, close and read/write random access disc files, the next step is to develop the technique. Although we are going to forsake BASIC altogether from now on it is very useful to know just how data is read and written from random access files.

To date only the BPUT and BGET have been used to write and read data. The action of these commands is straightforward enough – a single byte is sequentially operated on. But BASIC is supposed to be friendly, which is why it is also possible to use statements like PRINT# and INPUT# in relation to a random access file. Using these rather more familiar words a programmer can send strings and numbers to an open file and read them back at a later date without taking much notice of the fact that the data is coming from disc rather than the keyboard.

This could lead onto the concept of **devices,** but we are more concerned with just one device at the moment – a disc file. Using random access files from BASIC actually becomes more difficult than from machine code when using non-byte operands, because the pointer (as returned by PTR£) still gives us byte-orientated results, although the **records** which are being read and written are very unlikely to be so organised.

Obviously it is necessary to keep records of known length when using PRINT# and INPUT# to write and read data respectively. BASIC stores strings that are printed to an open file in a fairly peculiar order. First there is a byte containing zero (to indicate string data), then a length byte (no BBC Basic string can exceed 255 characters) and finally the characters of the string *in reverse order.*

Numbers are sent in their internal five-byte floating point format, with a preceeding byte containing &FF to indicate a 'real' number, unless an integer was sent in which case the first byte is &40 followed by the *four* bytes of the number.

This makes it rather easier for the Basic interpreter to make sense of what it is reading when subsequent INPUT statements are used, but of course it makes it a lot harder to play around with the PTR# statement later on. We need to know exactly what is on the disc in order to make any sense of what is being read.

Naturally enough these conmventions are forsaken when using random access files from machine code – there is no point in adding complications. We can however still read more than one byte from a file at a time using another operating system call – this time to OSGBPB, which also allows us to write more than one byte at a time. This operating system routine is actually rather more versatile than this, as it also allows the user to dictate whether the current value returned by PTR# or an entirely different pointer will be used when reading or writing. this is how it works:

The routine is called with the 6502's X and Y registers together pointing to an address of a control block that can be located anywhere in memory. This control block contains the channel number (the file 'handle'), the address of the data to be transferred (or the destination address in the case of a read), a value to be used as the sequential pointer and the number of bytes to be transferred. OSGBPB is basically an OSARGS call followed by multiple OSBPUT or OSBGET calls, but is considerably faster.

The contents of the A register on entry to the routine determine the action that is taken. the only ones that need concern us here are calls 1 and 2 (write bytes) and calls 3 and 4 (read bytes). Calls 1 and 3 use the pointer held in the control block when they are reading and writing, while calls 2 and 4 use the current pointer. Note that the sequential pointer in the control block – if used – is relative to the *start* of the file, *not* the current pointer, so putting 0 in here and using a call of 1 or 3 will take effect from the physical beginning of the file.

The control block is laid out like this:

XY+0 file handle

XY+1 address of data for a write, destination of data for a read. Lowest significant byte first.

XY+5 a four-byte value (lowest significant byte first) representing the number of bytes to be read or written.

XY+9 the four-byte sequential pointer (if used).

After the call, all used parts of the control block are updated to hold their new actual values. This routine is very useful for situations where the records in a random access file are of a length greater than one byte. There is no real limit to a record, except of course that which is imposed by the physical limits of the disc, but most operating systems, such as CP/M, settle on a pre-defined record size. This size is not easily alterable within the context of the operating system. Acornsoft's BCPL system for the BBC Micro does not itself support random access files, but a package is available that offers such a facility. In this system a record is defined as being 128 BCPL words long, which is 256 bytes – very conveniently the sector size of a BBC Micro disc. the package was written by the author and makes extensivew use of OSGBPB.

Now let's think of a useful application for a random access file and write it. To start this is written in BASIC to make it easier to understand initially, and then converted to 6502 machine code later on.

Suppose we have a file on disc that comprises a number of frames saved from Micronet. They will therefore be nothing more than a collection of Mode 7 screen images each &398 bytes long. We want a short routine that allows viewing of these frames at random. Easy. All that has to be done is to move the file sequential pointer to the start of that particular screen and load &398 bytes from there on into screen

```
10  ON ERROR GOTO 210
20  MODE 7
30  VDU 28,0,24,39,22
40  INPUT "Filename: "F$
50  S%=OPENIN(F$)
60  IF S%=0 THEN PRINT "No such file!";:GOTO 40
70  N%=(EXT#S%)/&398
80  REPEAT
90  CLS
100 INPUT "Screen number: "R%
110 IF R%>N% THEN UNTIL TRUE:GOTO 210
120 R%=(R%-1)*&398
130 REM set up control block at &70
140 ?&70=S%
150 !&71=&FFFF7C00:REM Make this &FFFF3C00 for model As
160 !&75=&398
170 !&79=R%
180 A%=3:X%=&70:Y%=0
190 CALL &FFD1
200 UNTIL FALSE
210CLOSE#S%
220 END
```

memory. Humans normally start counting from 1, so the record number entered will be from 1 up, whilst computers generally start from 0. So by taking the record number given, subtracting 1 and multiplying by &398, we are left with the correct value for the sequential pointer. We can then use OSGBPB to load it from disc. This process is repeated until a non-existant record is requested or until ESCAPE is pressed.

As the OSGBPB control block must always lie in the I/O processor's memory, this program will not work correctly with the very wonderful TUBE. Not that this is too important – Acorn made a big fuss about nothing. After all, all you have to do is switch your second processor off! This is the program:

This program will of course only work on a disc file containing the requisite Mode 7 screens already in the pre-defined format. If you don't have access to Micronet or similar Prestel and Teletext services, you'll have to redesign the program a little and think of another similar simple application of your own.

If you do have access to Micronet but do not know how to create a file of this format, save a number of frames onto disc all with similar names but sequentially numbered. Like *screen1, screen2, screen3,* etc. Save as many as you like but don't fill the disc up! All these frames can be joined into one large file with a program like this:

```
10  S%=OPENOUT("FRAMES")
20  FOR a%=1 TO 20:REM or
    however many frames you saved
30  A$="screen"+STR$(A%)
40  T%=OPENIN(A$)
50  REPEAT
60  BPUT#S%,BGET#T%
70  UNTIL EOF#T%
80  CLOSE#T%
90  NEXT A%
100 CLOSE£S%
```

The file for use with the program has now been created with the name "FRAMES", but of course the name could be whatever you choose.

If you are using the Micronet ROM software rather than the Version 4 cassette or disc software, change every occurence of &398 to &3C0, as the ROM saves frames in a slightly different format. Next month, it's all registers go as we convert everything to machine code with rather more exciting and useful examples.

Dianetics
The Evolution of a Science
by L. Ron Hubbard

PART 1

The optimum computing machine is a subject which many of us have studied. If you were building one, how would you design it?

First, the machine whould be able to compute with perfect accuracy on any problem in the Universe and produce answers which were always and invariably right.

Second, the computer would have to be swift, working much more quickly than the problem and process could be vocally articulated.

Third, the computer would have to be able to handle large numbers of variables and large numbers of problems simultaneously.

Fourth, the computer would have to be able to evaluate its own data and there would have to remain available within it not only a record of its former conclusions but the evaluations leading to those conclusions.

Fifth, the computer would have to be served by a memory bank of nearly infinite capacity in which it could store observational data, tentative conclusions which might serve future computations and the data in the bank would have to be available to the analytical portion of the computer in the smallest fractions of a second.

Sixth, the computer would have to be able to rearrange former conclusions or alter them in the light of new experience.

Seventh, the computer would not need an exterior program director but would be entirely self-determined about its programming guided only by the necessity-value of the solution which it itself would determine.

Eighth, the computer should be self-servicing anmd self-arming against present and future damage and would be able to estimate future damage.

Ninth, the computer should be served by perception by which it could determine necissity-value. The equipment should include means of contacting all desirable characteristics in the finite world. This would mean color-visio, tone-audio, odor, tactile and self perceptions – for without the last it could not properly service itself.

Tenth, the memory bank should store perceptions as perceived, consecutive with time received with the smallest possible time divisions between perceptions. It would then store in color-visio (moving), tone-audio (flowing), odor, tactile and self sensation, all of them cross-co-ordinated.

Eleventh, for the purposes of solutions, it would have to be able to create new situations and imagine new perceptions hitherto not perceived and should be able to conceive these to itself in terms of tone-audio, color-visio, odor, tactile and self sensation and should be able to file anything so conceived as imagined labeled memories.

Twelfth, its memory banks should not exhaust on inspection but should furnish to the central perceptor of the computer, without distortion, perfect copies of everything and anything in the banks in color-visio, tone-audio, odor, tactile and organic sensations.

Thirteenth, the entire machine should be portable.

There are other desirable characteristics but those listed above will do for the moment.

It might be somewhat astonishing, at first, to conceive of such a computer. But the fact is, the machine is in existence. There are about two billion of them in use today and many, many more billions have been made and used in the past.

In fact, you've got one. For we're dealing with the human mind.

The above is a generalisation of the optimum brain. The optimum brain, aside from the fact that it is not always capable of solving every problem in the Universe, basically works exactly like that. It should have color-visio (in motion), tone-audio (flowing), odor, tactile and organin memory recall. And it should have color-visio (in motion), tone-audio (flowing), odor, tactile and organic imagination, also recallable after imagining like any other memory. And it should be able to differenciate between actuality and imagination with precision. And it should be able to recall any perception, even the trivial, asleep and awake from the beginning of life to death. This is the optimum brain, that and much, much more. It should think with such swiftness that vocal pondering would be utterly unable to keep pace with a thousandth part of one computation. And, modified by viewpoint and educational data, it should be *always* right, its answers *never* wrong.

That is the brain you have, potentially. That is the brain which can be restored to you unless you have had some section of it removed. If it does not do these things, it is slightly out of adjustment.

It took a long time to arrive at the data that this was an optimum brain. In the beginning it was not realised that some people had color-visio – moving – recall, for instance, and that some did not. I had no idea that many people imagined, and knew they were imagining, in tone-audio, et cetera, and would have received with surprise that data that somebody could smell and taste last Thanksgiving's turkey when he recalled it.

Fifteen years ago, when the researches which culminated in Dianetics (Gr. *dianoua* thought) were started in earnest, no such high opinion of the human brain was held. In fact, the project was not begun to trace function and restore optimum operation, but to know the key to human behaviour and the code law which would reduce all knowledge.

My right to enter this field was an inquiring brain which had been trained in mathematics and engineering and which had a memory bank full of questions and far-flung observations.

It was the basic contention that the uman mind was a problem in engineering and that all knowledge would surrender to an engineering approach.

And another primary assumption was made:

All answers are basically simple.

As it stands today, the science of Dianetics and its results – which are as demonstrable as the proposition that water, at fifteen pounds per square inch and 212°F, boils – is an engineering science, built heuristically* on axioms**. It works. That is the only claim for Dianetics or chemistry. They may not be True. But they work and work invariably in the finite world.

When the problem had been shuffled around, in the beginning, and when questions had been formulated to be asked of the Universe at large, there was no conept of the optimum brain. Attention was fixed upon the *normal* brain. The *normal* brain was considered to be the optimum brain. Attempts were made, when work finally got around to the problem of the brain itself, to obtain results comparable with the normal mind. Minds became aberrated***. When restored they would be normal.

In fact in the beginning, it was not even certain that minds could be restored. All that was required was an answer to existence and the reasons minds aberrated.

In a lifetime of wandering around many strange things had been observed. The medicine man of the Goldi People of Manchuria, the shamans of North Borneo, Sioux medicine men, the cults of Los Angeles, and modern psychology. Amongst the people questioned about existence were a magician whose ancestors served in the court of Kublai Khan and a Hindu who could hypnotise cats. Dabbles had been made in mysticism, data had been studied from mythology to spiritualism. Odds and ends like these, countless odds and ends.

If you were constructing this science, where would you have started? Here were all the various cults and creeds and practices of a whole world to draw upon. Here were facts to a number which makes 10^{21} binary digits look small. If you were called upon to construct such a science and to come up with a workable answer, what would you have assumed, gone to observe, or computed?

Everybody and everything seemed to have a scrap of the answer. The cults of all the ages, of all the world seem, each one, to contain a fragment of the truth. How do we gather and assemble the fragments? Or do we give up this nearly impossible task and begin postulating our own answers?

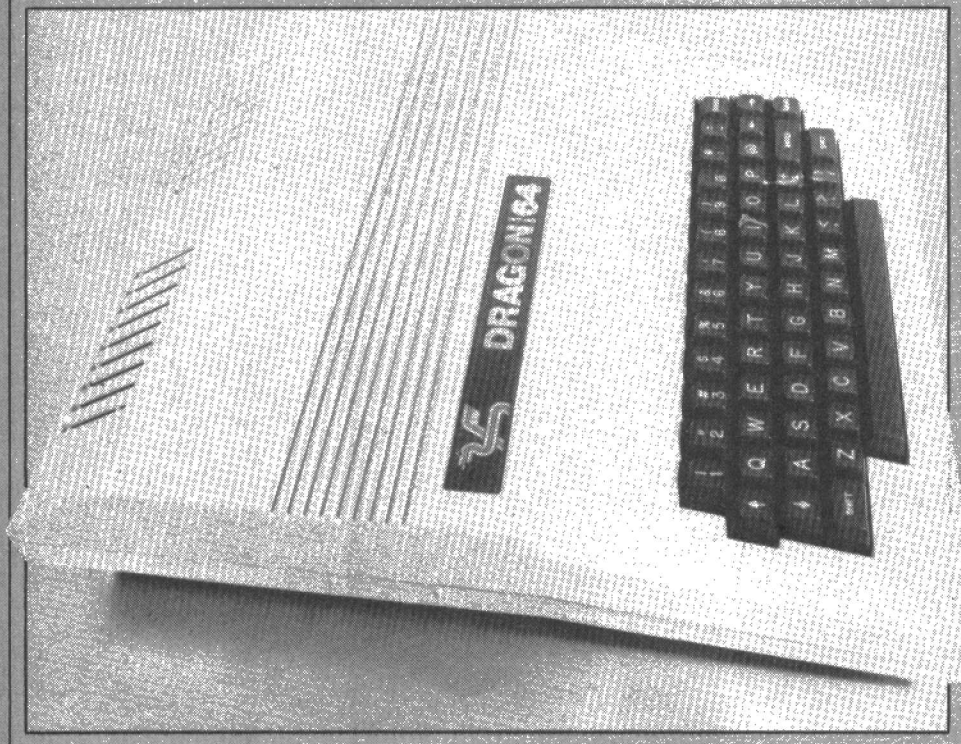
Well, this is the story of how Dianetics was built. This, at least, was the approach made to the problem. Dianetics works, which is what an engineer asks, and it works all the time, which is what nature demands of the engineer.

[Part 2 Next Month]

Heuristic: serving to guide, discover or reveal. **Axiom*: a proposition regarded as a self-evident truth. ***Aberrated*: departed from rationality, deranged.

OS9 AND THE DRAGON

Mike James continues his examination of OS9 and in particular how the Dragon's hardware complements the operating system.



The two way relationship between a computer's operating system software and the hardware on which it is to be run is crucial to the overall performance obtained. In the July issue of *E&CM* the basic ideas of what an operating system should do to improve the quality of the services offered by any computer were described yet the question of how the hardware of a computer can make the task of the operating system easier were avoided. This month we turn our attention to Dragon Data's implementation of OS9, the emphasis initially being on the hardware aspects of a successful OS and then on the applications software essential to make the entire system useful.

OS9 and the Dragon

The title of this section sounds like an adventure game and in a sense that is not far from the truth. The Dragon 64 is essentially a Dragon 32 with 64K of RAM, a serial port and a slightly better keyboard. This configuration compares favourably with the minimum hardware requirements to run OS9 – 6809 CPU, 24K of RAM, 4K of ROM and a VDU (which can be memory mapped). However, being realistic, the Dragon's 64K of memory represents the minimum convenient amount of memory for an operating system such as OS9. Just after OS9 has been loaded the command MFREE reports that there are 169 pages, or just over 42K, of RAM free. This is a very reasonable amount of free memory when compared with other 'single tasking' computers in the same range as the Dragon but it is not really a great deal of memory to share out between a number of different tasks under the 'multi-tasking' OS9. With only 42K of RAM you can forget any ideas that you might have had about running, say, Pascal and BASIC at the same time! However it is worth pointing out that all OS9 system software such as the BASIC interpreter are 're-enterant'. This means that a single copy of the BASIC interpreter can be shared between a number of BASIC programs with an obvious saving in memory space. Working out just what it is feasible to run as concurrent tasks under OS9 on the Dragon is quite a difficult business and doesn't just depend on there being enough free memory; it also depends on exactly what the various tasks are doing, but more of this later.

If you are not interested in multi-tasking, then perhaps the most important question to ask of OS9 on the Dragon is how fast it accesses files etc. The answer is that it is slow in comparison with simpler single tasking operating systems such as Flex or CP/M. For example, when you ask for a directory of what is on a disc there is much whirring and clunking of the disk drive before the directory appears on the screen. This delay (using 5 inch drives) is of the order of a couple of seconds and is not unreasonable but it is longer than it takes a single tasking operating system to do the same job and this is something that you might feel is important in selecting a system.

The Dragon 64 has the same set of display modes as the Dragon 32. This is good news if you are interested in high resolution graphics in colour but unfortunate if you want a full 80 column by 24 line text display with both upper and lower case. The only hardware generated text screen that either the Dragon 32 or the 64 provide is a very inadequate 32 column by 16 line, upper case only display. While the 32 column 16 line part of the specification is almost acceptable, the fact that it is only upper case severely restricts its use in serious applications. However, this is something that Dragon Data have realised and done something to put right with a software generated 51 column by 24 line upper and lower case text display. This works by using one of the high resolution graphics modes to draw the characters on the screen. The result is not as clear a display as you would find on most VDU's or professional systems but it is a great deal better than not having lower case at all! Of course, as it uses a high resolution screen, there is a penalty of 6K of valuable RAM occupied by text output but memory critical applications can always go back to the upper case only screen.

"The Dragon's hardware generated text screen is inadequate for many uses".

The only other feature of the Dragon 64 that has any relevance for OS9 is the presence of a serial interface that can be used to connect a VDU or printer and a centronics standard parallel printer port. Device drivers for both of these ports are provided as standard.

The disk bottleneck

There are systems that will enable you to run OS9 in such a way that it will support a number of users and a number of tasks

without slowing to a halt – the Dragon 64 isn't one of these systems! To put this statement into perspective it is necessary to add that these 'super' systems often cost ten times the price of the Dragon 64 including its 5 inch disk drives. In other words the Dragon is excellent value but it is hardly state of the art hardware any more. It is worth examining in more detail the reason why the Dragon is not as fast as it could be as the reasons help to outline the limitations of the machine.

Sharing CPU power

If you recall July's introduction to operating systems you will know that the key to any advanced opoerating syustem is the sharing of the processing power of the CPU. By switching the 'attention' of the CPU between a number of programs or tasks then there is more hope of using the other resources of the machine, such as the memory, printer and disk drives, efficiently. There is also the possibility that the total time that a number of programs takes to run may be reduced by running them together! This may sound paradoxical (or magical depending on your point of view!) but notice that it is the TOTAL running time that might be reduced and not the time that each program takes to complete its task. is a great advantage to any user.

This argument makes it appear that running more than one task at a time will always bring an improvement in system performance – the truth is rather different. The example of printing a text file while editing another was chosen because both tasks spend most of their time waiting for things to happen in the outside world – and in this case the tasks can be run together to some advantage. Now consider running two programs together that do hardly any input/output but spend their whole time calculating – that is, using the CPU. If each program takes an hour to run then running them one after the other would take two hours but, unlike the previous example, running them concurrently would take more than two hours. The reason for this is that while the CPU was running the first program, the second program would just sit there doing nothing and vice-versa. Thus it would still take two hours for both programs to finish and in fact it would take rather more than this because of the extra time needed to switch the attention of the CPU between the two!

In short, to make the best use of a multi-tasking operating system, you have to choose a 'suitable' mix of programs to run together. This is the reason why it is not

"To the uninitiated, OS9's multi-tasking can give rise to seemingly magical effects".

For example, suppose that you want to edit a file and print out the results of an earlier text editing session. Let's also suppose that you are going to use a slow, 10 character per second, high quality printer and then the total printing time could run into more than an hour. If your proposed text editing session is also likely to take more than an hour, then the total time to complete both tasks using a conventional single tasking system is more than two hours. However, neither the printing nor the editing task is using much in the way of the computer's resources for the two hour period. The printing task spends most of its time waiting for the slow printer to be ready to accept another character while the editing task is occupied with waiting for the ponderous typist to enter the next character. Remember, from the point of view of a computer, executing one instruction every millionth of a second ALL printers and ALL typists are extremely slow. Now consider this example run under OS9 as a pair of concurrent tasks. Both the printing program and the text editor can be loaded into memory at the same time and the CPU now divides its working time between them. Neither of the tasks run any slower because of this division of the CPU's attention because both of them still spend most of their 'active life' waiting for things on the outside world. The result is that the file is printed and the text is edited in the same one hour period (providing, that is, the printer is in a different room from the typist!) Thus running two tasks together can reduce that total running time and this easy to say how many tasks can be run on the Dragon at the same time – it all depends on what the tasks are. For example, a printing task is always worth running concurrently with something else, as is any data entry program but, apart from these two generalisations, the only way to see what happens when tasks are run at the same time is to actually try it.

Disk Operation

There is a particular problem with multi-tasking caused by the way that the Dragon's disks work. The Dragon disk interface uses a fairly standard disk controller chip that needs a lot of attention from the CPU during data transfers. On the face of it sending data to a disk controller looks a lot like sending data to a printer interface and, as such, you might expect that programs that spend much of their time handling disk files would be candidates for running concurrently with other programs. This is unfortunately not the case – disk is a CPU intensive priority device and not only tends to soak up a great deal of the CPU's time but it does so in a very anti-social way. Data transfers to and from disk are handled in chunks, usually 256 bytes at a time, called sectors. Unlike a printer, you cannot send single bytes to a disk as and when you like – once you have started sending the bytes that constitute a sector you have to keep on until they are all written out to disk. In this sense a disk operation is a 'priority' operation that must be performed at the time that the disk interface demands. This means that while a disk operation is in progress nothing else can happen – the keyboard will not be scanned, text will not be printed to the screen, no other processing of any sort can take place. You can see this effect very easily if you try to run the OS9 disk check program DCHECK concurrently. The result is that while it is running the

```
Volume - 'DRAGON DISK' on device /d0
$005A bytes in allocation map
1 sector per cluster
$0002D0 total sectors on media
Sector $000012 is start of root directory FD
$000A sectors used for id, allocation map and root directory
Building allocation map work file...
Checking allocation map file...
Cluster $00000A in allocation map but not in file structure
Cluster $00000B in allocation map but not in file structure
Cluster $00000C in allocation map but not in file structure
```

Print out obtained from OS9's DCHECK command.

```
Directory of cmds   00:00:15
RUNB        attr        backup      makdir      mdir        merge
binex       build       cmp         mfree       os9gen      printerr
cobbler     copy        date        procs       pwd         pxd
dcheck      rmsnew      del         rename      save        setime
deldir      dir         display     shell       sleep       tee
dsave       dump        echo        tmode       tsmon       unlink
rms         exbin       format      verify      xmode       go51
free        ident       link        acia51      index       report
list        load        login       rmscopy     rms.trm     stylo
```

Directory of OS9 commands file.

keyboard becomes almost unresponsive and everything grinds to a halt. The problem cannot be solved by giving the DCHECK program a low priority (or conversely, the shell a high priority) because once a disk operation has started it keeps the CPU busy until it is finished and this is the reason for the break down of the otherwise 'fair' system of sharing the CPU. The only real solution to this problem is to improve the hardware. Most of the more expensive systems that run OS9 include sophisticated disk interfaces using DMA (Direct Memory Access) chips. The idea here is that a DMA chip can handle all of the data transfer to and from the disk without bothering the CPU with such a menial job.

```
Directory of .   12:00:27
OS9Boot     CMDS        SYS
DEFS        startup     RUN32
RUN51       dcheck060
```

Directory of OS9 master disc.

Indeed, in a well designed system, the DMA chip will access the memory during times when the CPU isn't using it, thus ensuring that the CPU is free to process another task while the disk in in use. These clever hardware touches can also be applied to other I/O devices as well as disk to produce extremely powerful systems. For example, a 6809 system built by GIMIX includes a small CPU as part of a multi-port serial interface so that most character input (and indeed line editing) can be done without the help of the main CPU. This all sounds very exciting and it does tend to highlight the hardware inadequacies of the Dragon 64, but there comes a point in building such sophisticated machines, capable of handling a number of users, that the question of whether the whole thing might have been better handled by buying a number of smaller machines! (Indeed it is

"There is a wealth of OS9 software available".

an interesting and important question to ask what the future of microcomputing holds – small powerful personal machines or small but even more powerful multi-user/multi-tasking machines shared between a number of users – but this is not the place to explore the possible answers!)

OS9 software

There is so much OS9 software that a complete review of each product would fill *E&CM* for a couple of months! The best that can be achieved in a reasonable amount of space is to list what Dragon Data have issued along with comments on the more important items.

BASIC 09 £59.95
BASIC09 is a better BASIC than you will find on almost any micro. For example, it includes all of the facilities that have caused BBC BASIC to be admired and a great deal more. In many ways it is better to describe BASIC09 as a cross between standard BASIC and Pascal. It should be sufficient to say that many people have switched to OS9 just so that they can use BASIC09!

Editor, Assembler and Debug £49.95
There is not much to say about this collection of programs apart from the obvious fact that they are essential if you want to develop machine code programs under OS9. The editor is more properly described as a 'macro' editor, meaning that it is possible to define new editing commands using the existing set.

Pascal £79.95
OS9 Pascal is remarkable in that it is a full implementation of Pascal as a compiler that can generate either P-code or native 6809 machine code. This means that it is both fast and flexible and well worth the £79.95. It also has the ability to treat disk storage as 'virtual' memory thus making it possible to run programs that take more than the available RAM!

C Compiler £79.95
C is not a very well known language at the moment but is seems set to become the system implementation language of the future. The reason for this is simply that it is the 'standard' language for any UNIX system.

Record Management System £54.95
A very versatile data base management program that can work with Stylograph to produce mailing lists etc.

Cash and VAT £49.95
Fairly standard but good quality VAT and cash accounting program – excellent value!

Dynacalc £59.95
This is one of the lowest cost, full function spreadsheet calculators available.

Stylograph, Spellcheck and Mailmerge £79.95
For the cost of a text processor alone you can buy Stylograph, a friendly screen text processor; Spellcheck, an excellent spelling check program; and Mailmerge, a mailing list manager. These three programs make OS9 excellent for any text processing application.

Appropriate technology?

OS9 is a sophisticated 'state of the art' operating system but it is now difficult to claim that the Dragon is 'state of the art' hardware. In many ways this mis-match is totally irrelevent – the 6809 is still one of the most powerful microprocessors available and it is well suited to OS9 and vice versa. Using OS9 with the Dragon certainly turns it into a different sort of computer from its original Dragon 32 fun-loving game-playing image. The trouble is that it is difficult to see what its new role in life is going to be. With Dragon Data's recent announcement of the "Professional", essentially a Dragon 64 with built-in 3 inch disks, they clearly see the future of the Dragon as a low cost but powerful business system. Also with Microware's recent announcement that OS9 is now available for 68000 based systems it is easy to see that there is a future for OS9 in 'serious applications'. However, looking at the team formed by the Dragon 64 and OS9, it is difficult not to think that OS9 does more for the Dragon than the Dragon does for OS9. The hardware is quite good, but it is possible to see how it could be made so much better and the prospect of OS9 running on low cost but purpose built hardware is extremely tantalising.

DIGITAL FREQUENCY METER

Transform the Spectrum into a digital frequency meter with the latest add-on from Robert Penfold. The unit is low-cost and involves the minimum software overhead.

This Spectrum add-on converts the machine into a digital frequency meter for audio frequency use. The cost of the unit is comparable to a simple analogue frequency meter, but it has the advantage of greater accuracy. Nor does it require any difficult calibration or setting up once completed. the unit covers a frequency range of 0 to 65535 Hertz with a resolution of 1 Hertz, and the input sensitivity is approximately 5 millivolts RMS into 900k. Input protection circuitry enables high input levels of more than 100 volts peak to peak to be tolerated.

System operation

In principle, a digital frequency meter is the most simple form of frequency measuring equipment. the basic arrangement used is shown in the block diagram of **Figure 1.**

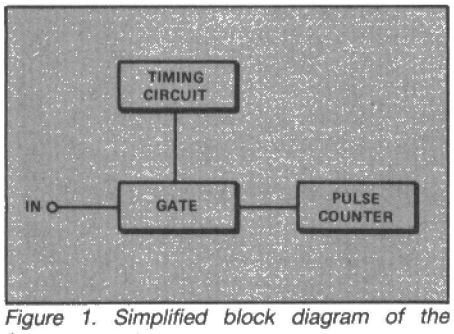


Figure 1. Simplified block diagram of the frequency meter.

The pulse counter circuit is initially set at zero, and the timing circuit then activates the gate for a certain length of time. The number of input pulses that are registered for a given gate period obviously depend on the frequency of the input signal, and are proportional to the frequency of this signal. The resolution of the unit depends on the length of the gate pulse, and the longer the pulse length then the greater the resolution. Many frequency meters for radio frequency use have quite short gate periods (typically about 10ms) but, for audio frequency use, a fairly long gate period of about one second is needed in order to give good resolution and accuracy. A one second gate pulse is a convenient length to use in practice since it enables the reading to be updated quite frequently; it gives an adequate resolution of 1 Hertz; and the number in the counter at the end of the gate period is equal to the input frequency (in Hertz). A one second gate period has therefore been chosen for this design.

Frequency meter applications

The Spectrum Frequency Meter is ideally suited for use in conjunction with a low cost AF signal generator. Low cost signal generators are often capable of producing outputs that are stable in frequency but the frequency output often bears little relationship to that indicated on the equipment's own scale.

The frequency meter allows the output to be set to the exact frequency required. This is essential if the results obtained from using the signal generator are to be reliable.

A common application would be to measure the frequency response of an amplification system. The generator would provide the input while the level at the output would be monitored with an audio millivoltmeter. Plotting a graph of frequency vs output will provide a graphic guide to the performance of the systems. The action of tone controls and filters could be examined in this fashion.

Another application using a similar technique would be to examine the frequency response of a room.

Of course, a practical frequency meter is more complex than the block diagram of **Figure 1** would suggest, since some control logic is required, and the timer circuit must be a crystal controlled type if good accuracy and consistent results are to be obtained. The use of a crystal based timer circuit also eliminates the need to calibrate the finished unit as it automatically has a very high degree of accuracy.

"A low cost design with an up-market specification".

Figure 2 shows the block diagram for this frequency meter adaptor, which is based on the Z80A CTC. This is the counter timer chip in the Z80A series of peripheral devices, and it contains four 8-bit counter timers. There are two basic ways of using the timers. In the counter mode pulses that are fed to an input pin are counted, and all the timers are down counters. There is an output which pulses high each time a zero count is reached. In the second mode clock pulses (normally supplied by the system clock) are counted. This second mode is the "timer" mode, and it is normally used to provide output pulses at regular intervals on the zero count output of the counter, or to generate interrupts at regular intervals. In the timer mode a pre-scaler which divides by either 16 or 256 has to be used ahead of the 8-bit counter, but the pre-scaler is not available in the counter mode.

In this circuit one of the 8-bit counters is used in the timer mode and, together with another counter which is used in the counter mode, it acts as a divider which forms part of the circuit that generates the gate pulse. The Spectrum has a clock output, but this is not a regular signal and it is not used as the timebase for this design. Instead a separate oscillator using an inexpensive 4MHz crystal is included in the frequency meter adaptor and acts as the timebase generator. This signal is first divided by 16 in the channel 0 pre-scaler, after which it is divided by 250 in the 8-bit counter of channel 0. The counter operates

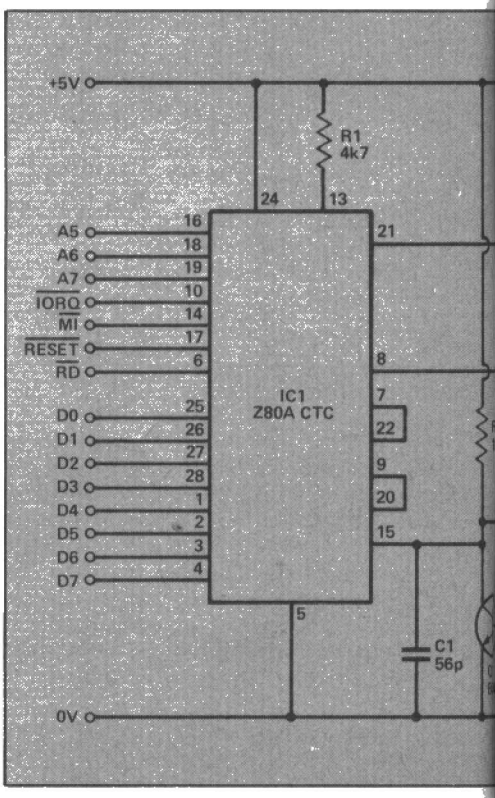


Figure 3. Full circuit diagram of the digital frequency

SPECTRUM ADD-ON

in the timer mode when a number to be loaded in. Input pulses can cause the counter to count downwards until zero is reached, at which point it is automatically re-loaded to its starting value, an output pulse is produced, and the cycle is repeated indefinitely. Therefore, in order to obtain the required divide by 250 action it is merely necessary to use a software routine to load 250 into the 8-bit counter.

The output pulses from channel 0 are fed to the input of the channel 1 counter. This is used to give a further division by 250 to give an output frequency of 4HZ. This signal is then divided by a discrete divide by 8 circuit which produces an output frequency of 0.3HZ. This gives an output signal that is low for one second, high for one second, and so on, giving the required one second gate period, at one second intervals. The frequency reading is therefore updated every two seconds.

In most cases the input signal will not be at standard logic levels, and will not even be a pulse of squarewave type. Some interfacing circuitry between the input and the gate is therefore needed.

A simple clipping circuit provides overload protection, and this is followed by a buffer amplifier which gives the unit a high input impedance so that minimal loading is placed on the circuit under investigation. The next stage is an amplifier which boosts the sensitivity by over 40dB (100 times) and this is followed by a trigger circuit. The latter gives an output at ordinary logic levels provided the input level is sufficient to drive it properly. If the input signal is inadequate there is no output signal from the trigger circuit, and this helps to avoid misleading readings with marginal input levels.

The 16-bit counter is formed by the two

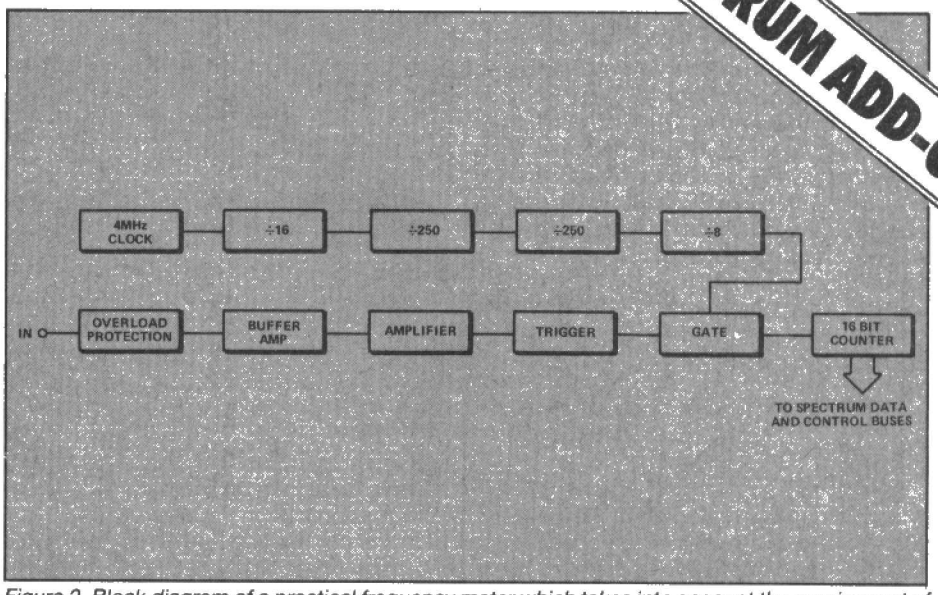


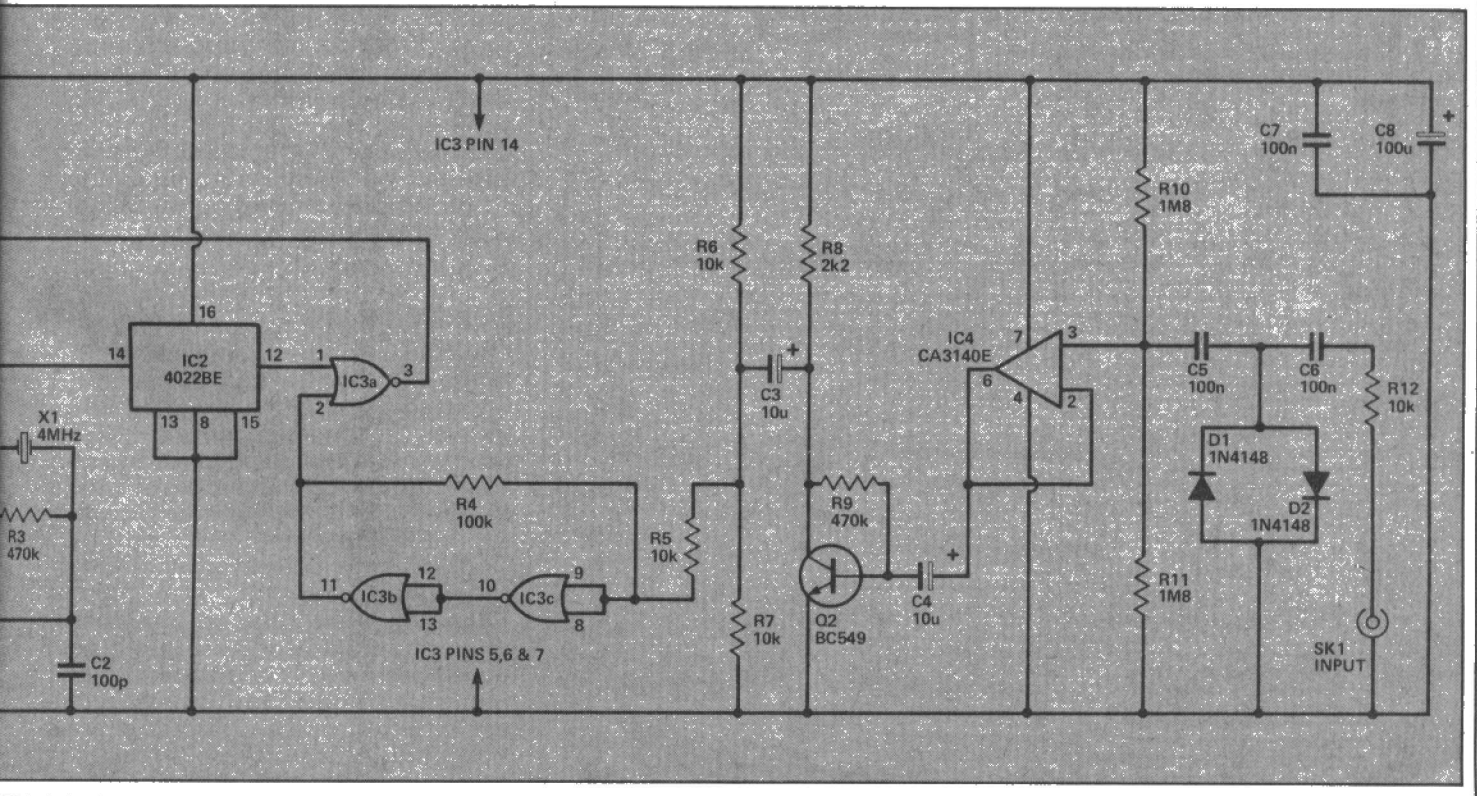
Figure 2. Block diagram of a practical frequency meter which takes into account the requirement of signal pre-conditioning and sophisticated gating circuitry to ensure overall accuracy.

8-bit counters of channels 2 and 3, which are connected so that channel 3 counts the channel 2 zero-count output pulses. Channel 2 therefore provides the low byte while channel 3 provides the high byte of the result.

On the face of it there is a flaw in this arrangement, in that there is no control circuit to ensure that the 16-bit counter is read and then reset to 65535 at suitable times. In fact the computer effectively provides the control circuit, although it is really software rather than hardware that handles this. The basic technique is to read the counter, and then take a second reading about a quarter of a second later. If the readings are different the counter is assumed to still be counting, and a new set of readings are taken. When two identical readings are obtained the counter ceases operation, and after some mathematical manipulation the result is displayed. The counter is then reset to 65535 ready for the next reading to be taken. This system works very reliably in practice, and due to the fairly long gate period a BASIC routine is fast enough to handle things properly.

Circuit operation

As can be seen from the circuit diagram of **Figure 3,** the Z80A CTC (IC1) interfaces direct to the expansion bus of the Spectrum without the need for an address decoder. IC1 has inputs for the IORQ, M1, RESET, and RD control lines of the computer. In common with other Z80 peripherals, no WR input is provided and the WR



This is built around the Z80 CTC which is the work horse of the design.

signal is generated within the device by an absence of an active RD signal. Address line A5 of the Spectrum is high whenever the computer is performing an internal input/output operation. This line can therefore be connected to the negative chip select input of IC1, and by using IN/OUT addresses that take this line low the device can be activated. Address lines A6 and A7 are not used by the Spectrum for internal input/output operations, but to control the register select inputs of IC1. This places the four channels of the device at the following addresses:

Channel 0	65311
Channel 1	65439
Channel 2	65375
Channel 3	65503

As only partial decoding of the address bus is used it is possible to access each channel using numerous addresses, but it is advisable to use those listed above as spurious operation of the Spectrum's hardware will then be avoided.

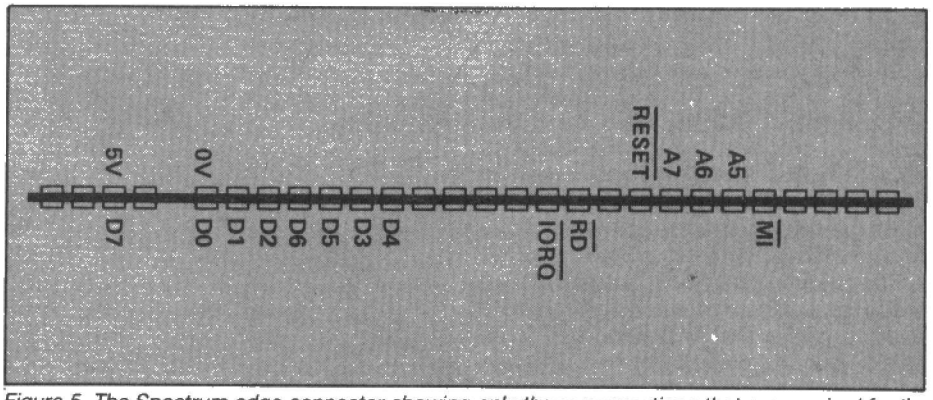


Figure 5. The Spectrum edge connector showing only those connections that are required for the DPM project.

In this application the interrupt capability of IC1 is not needed and R1 is used to tie the daisy-chain interrupt input to the positive supply rail. The 4MHz clock signal is generated by a conventional single transistor crystal oscillator based on Tr1. No facility to trim the operating frequency to precisely 4MHz has been included as any error here will be too small to affect the accuracy of the unit.

The 4Hz output of the channel 1 timer is coupled to the clock input of IC2 which is a CMOS 4022BE 1 of 8 decoder. In this application it is only the "carry out" output at pin 12 that is used, and this provides a straightforward divide by eight action. IC3a is the signal gate, and this is actually one section of a CMOS 4011BE quad 2 input NAND gate. This allows the signal applied to pin 2 to produce an output at pin 3 when the output of IC2 is high. The output from IC3a is coupled direct to the input of the channel 2 counter.

Two of the gates in IC3 (IC3b and IC3c) are connected as inverters and used in a simple Schmitt Trigger circuit, but the fourth gate is left unused. The amplifier stage has Tr2 in a standard high gain common emitter amplifier configuration. The buffer stage uses IC4 as a unity gain non-inverting amplifier which is biased by R10 and R11. These have high values so that a suitably large input impedance is obtained. The overload protection circuit is just a simple diode clipping circuit using D1, D2, and R12.

Only a 5 volt supply is required by the circuit, and the Spectrum is well able to supply this at the required current drain of about 100 milliamps.

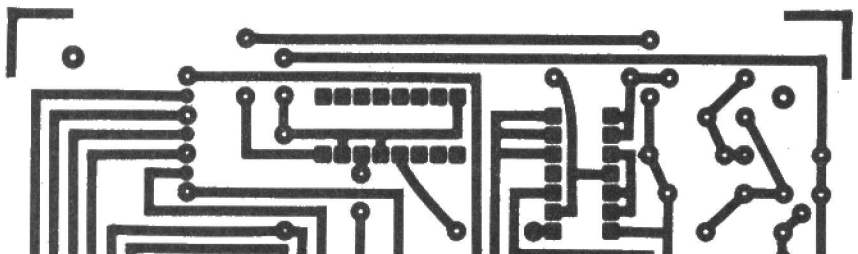
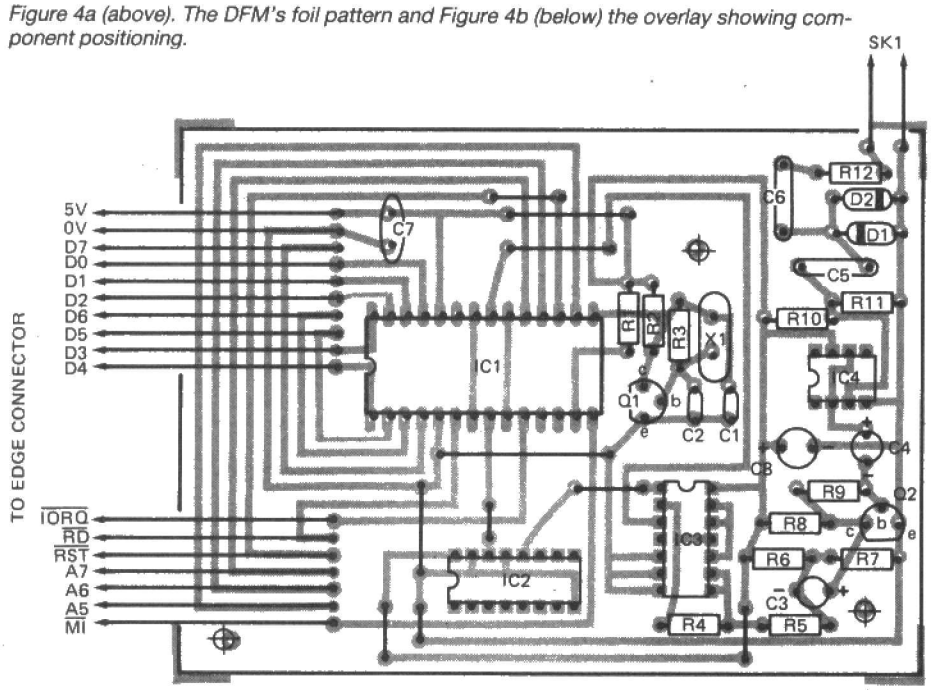


Figure 4a (above). The DFM's foil pattern and Figure 4b (below) the overlay showing component positioning.



Construction

The printed circuit design is reproduced in **Figure 4.** It is strongly recommended that integrated circuit holders should be used since all the integrated circuits are MOS types which are vulnerable to damage by high voltage static charges. Leave these devices in their protective packaging until the unit is otherwise finished, and then handle them as little as possible when plugging them into circuit.

The 4MHz crystal is also a delicate component, but in this case it is a matter of taking care to ensure that it does not overheat when soldered to the board. Provided each joint is completed within a second or two there is no real danger of seriously damaging this component. A plug-in crystal with a suitable holder could be used, but off the

shelf 4MHz crystals seem to be of the HC-18U type which have leadout wires rather than pins.

The printed circuit board is connected to the expansion bus of the Spectrum via a piece of 17-way ribbon cable about 0.5 metres or so in length. This is fitted with the now readily available Spectrum type 2 by 28-way edge connector at the computer end, and the other end connects direct to the printed circuit board. **Figure 5** gives wiring details for the edge connector, and the connections on the circuit board have been arranged so that they match up well with these connections. Even so, take great care to ensure that the board is wired to the edge connector correctly.

In use

Connect the unit to the computer before switching on. Once switched on the computer should function normally – switch off at once and re-check the unit if normal operation cannot be obtained. The following short program is all that is needed to make the equipment function as a frequency meter.

LISTING 1

```
10  OUT 65311,5
20  OUT 65311,250
30  OUT 65375,71
40  OUT 65375,250
50  OUT 65439,71
60  OUT 65439,255
70  OUT 65503,71
80  OUT 65503,255
90  LET x=IN 65439
100 LEY y=IN 65503
110 IF x=255 AND y=255 THEN
    GO TO 90
120 PAUSE 15
130 LET a=IN 65439
140 LET b=IN 65503
150 IF a=x AND b=y THEN
    GO TO 170
160 GO TO 90
170 LET z=y*256+x
180 PRINT AT 11,12;"°  "
190 PRINT AT 11,12;65535-z
200 GO TO 50
```

There is not sufficient space available here to permit a detailed description of the Z80A CTC and programming the device. However, the basic scheme of things is to first write a control number to the control register, and then write the required number to the eight bit counter. Reading a channel always reads the counter and not the control register incidentally. Thus each channel occupies only one address. A control number of 5 is used to set channel 0 in the mode where clock pulses are counter via the divide by 16 pre-scaler, and the other channels all have a control number of 71 which sets them to the mode where pulses on their input terminal are counted.

PARTS LIST

Resistors (¼W 5%)

R1	4k7
R2	1k
R3,9	470k
R4	100k
R5,6,7,12	10k
R8	2k2
R10,11	1M8

Capacitors

C1	56pF ceramic plate
C2	100pF ceramic plate
C3,4	10uF 25V radial elect
C5,6	100nF polyester
C7	100nF ceramic
C8	100uF 10V radial elect

Semiconductors

IC1	Z80A CTC
IC2	4022BE
IC3	4011BE
IC4	CA3140E
Tr1,2	BC549
D1,2	IN4148

Miscellaneous

SK1	3.5mm jack
X1	4MHz HC-18U crystal

Metal case about 133 by 102 by 38 millimetres; Printed circuit board; Spectrum type 2 by 28 way 0.1 inch edge connector; 28 pin DIL IC socket; 16 pin DIL IC socket; 14 pin DIL IC socket; 8 pin DIL IC socket; 17 way ribbon cable, wire, fixings, etc.

MSX HARDWARE HIGHLIGHTS

In previous months Gary Evans has looked at various aspects of the MSX specification. This month the emphasis is on a detailed examination of a typical MSX computer's hardware.

The major features of the MSX standard can be summed up in just a couple of sentences. Take a Z80 MPU, a Texas VDP and GI PSG, throw in 32K or ROM-based MSX BASIC and you will have a thumb nail sketch of an MSX machine. The July issue of *E&CM* provided just such an overview of the MSX hardware although it ran to slightly more than three sentences! Since preparing that feature *E&CM* has obtained a copy of the full MSX specification which reveals that there are quite a number of important aspects of the hardware architecture that were not fully explained in our earlier feature.

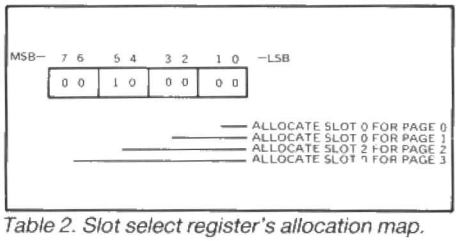


Table 2. Slot select register's allocation map.

MSX A to Z

The memory map of **Table 1** reveals that the MSX standard introduces the concept of a 'slot'. More on this later, for now it's sufficient to note that slot 0 is designated the 'system slot'.

The BASIC ROM (together with the OS) occupies memory locations from 0000h to 7FFFh leaving the upper half of the slot available for the system RAM. MSX BASIC will use the largest available RAM area that is installed from FFFFh to 8000h for its system working RAM. A minimum MSX configuration will consist of just 8K of RAM although it can be seen that the system treats address space in units of

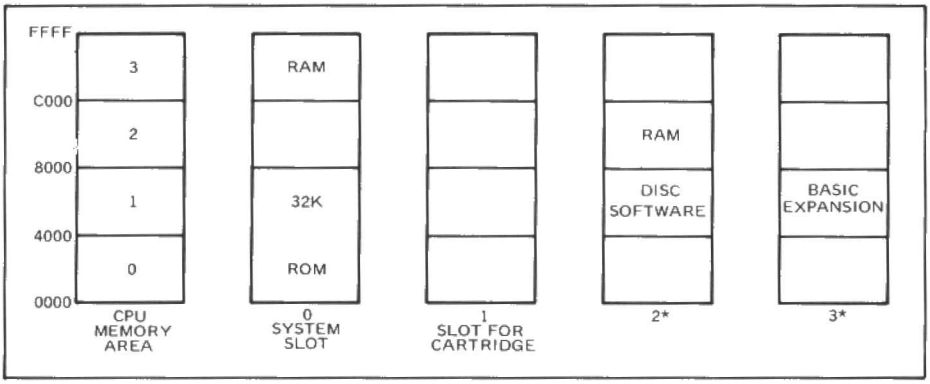


Table 1. MSX specification memory map.

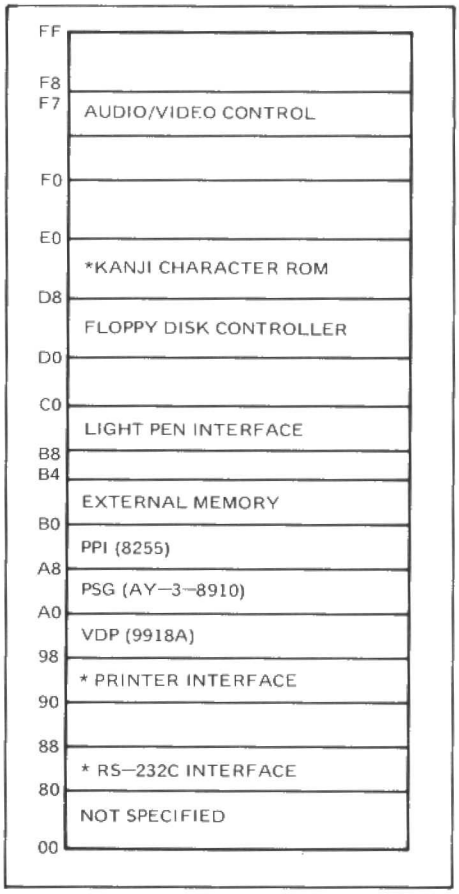
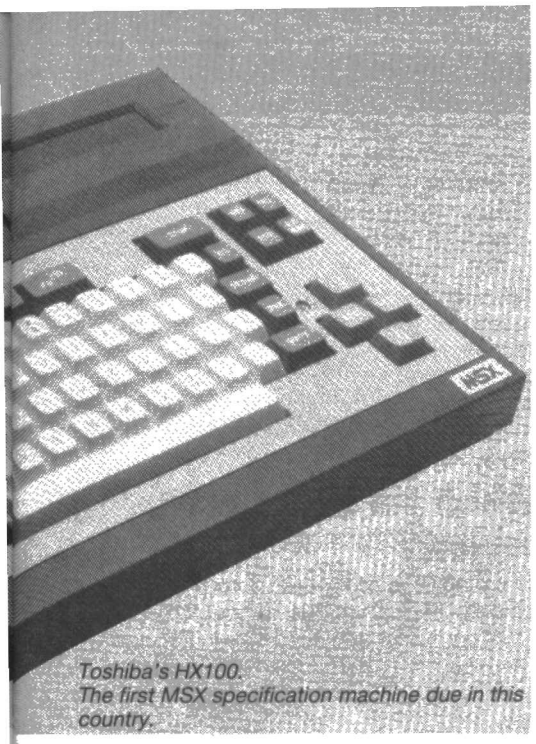


Table 3. I/O memory map.

MODE		RES	SIZE	No.	COLOUR	SPRITE	NUMBER OF CHARACTERS
GRAPHIC I	LSI SPEC	256 x 192	8 x 8	256	16 COLOURS	YES	32 x 24
	SUGGESTED VALUE	240 x 192					29 x 24
GRAPHIC II	LSI SPEC	256 x 192	8 x 8	768	16 COLOURS	YES	32 x 24
	SUGGESTED VALUE	240 x 192					29 x 24
MULTI-COLOUR	LSI SPEC	64x48blk	4 x 4 /BLOCK	—	16 COLOURS	NO	32 x 24
	SUGGESTED VALUE	64x40blk					29 x 24
TEXT	LSI SPEC	256 x 192	8 x 6	256	2 COLOURS OUT OF 16 COLOURS	YES	40 x 24
	SUGGESTED VALUE	240 x 192					39 x 24

Table 4. Full specification of the VDP various modes.



Toshiba's HX100.
The first MSX specification machine due in this country.

16K and adding a 16K memory expansion cartridge will only add an extra 8K of memory. The top 8K of RAM will simply be in parallel with that of the computer.

Returning to the concept of a 'slot', the memory map shows that the basic specification calls for four slots so that the total memory space can be expanded to 256K (4 x 64K). Each slot in turn can be expanded to support another four giving a total of 16 slots that will support up to 1 Mbyte of address locations.

The slots are under the control of the slot select register which is port A of the 8255 PIA and **Table 2** shows the bit pattern that would allocate page 0 and 1 from slot 0, page 2 from slot 2 and page 3 from slot 0. Note that the physical memory is tied to a specific page and thus while it is possible to allocate page 2 of slot 3 as part of the map it is not possible to assign page 3 of slot 3 to page 0 of CPU memory space.

"MSX micros – not to be dismissed as mere games machines".

A minimum system must have two slots, one for the system and the other for the cartridge. The MSX specification sheet hints at the fact that the choice of the word 'slot' to describe the way in which memory is divided may led to confusion. Thus, while a slot allocated to a cartridge will be a slot both physically and conceptually, the system slot has no physical manifestation.

An eye for I/O

Table 3 shows how the 256 I/O locations of the Z80 processor are allocated within an MSX machine. We shall deal with the major aspects of this map in turn, starting with the RS232 port at 80h to 88h. This is based around an 8251 communications interface chip and an 8253 programmable interval timer IC. The port assignments within the RS232 area of the I/O map are as follows:

80h	R/W 8251 data port
81h	8251 command status port
82h	R baud rate setting switches
83h	R configuration setting switches
84h	W interrupt mask register
83h	R/W 8253 counter 0
85h	R/W 8253 counter 1
86h	R/W 8253 counter 2
87h	W 8253 mode register

PIN No.	NAME	I/O	PIN No.	NAME	I/O
1	$\overline{CS1}$	O	2	$\overline{CS2}$	O
3	$\overline{CS12}$	O	4	$\overline{SLTSL}$	O
5	RESERVED	—	6	$\overline{RFSH}$	O
7	$\overline{WAIT}$	I	8	$\overline{INT}$	I
9	$\overline{M1}$	O	10	$\overline{BUSDIR}$	I
11	$\overline{IORQ}$	O	12	$\overline{MERQ}$	O
13	$\overline{WR}$	O	14	$\overline{RD}$	O
15	$\overline{RESET}$	O	16	RESERVED	—
17	A9	O	18	A15	O
19	A11	O	20	A10	O
21	A7	O	22	A6	O
23	A12	O	24	A8	O
25	A14	O	26	A13	O
27	A1	O	28	A0	O
29	A3	O	30	A2	O
31	A5	O	32	A4	O
33	D1	I/O	34	D0	I/O
35	D3	I/O	36	D2	I/O
37	D5	I/O	38	D4	I/O
39	D7	I/O	40	D6	I/O
41	GND	—	42	CLOCK	O
43	GND	—	44	SW1	—
45	+5V	—	46	SW2	—
47	+5V	—	48	+12V	—
49	SOUNDIN	I	50	−12V	—

Figure 1. Cartridge slot specifications.

PIN No.	SIGNAL NAME
1	PSTB
2	PDB0
3	PDB1
4	PDB2
5	PDB3
6	PDB4
7	PDB5
8	PDB6
9	PDB7
10	NC
11	BUSY
12	NC
13	NC
14	GND

Figure 2. Printer port pin out.

PIN No.	SIGNAL NAME	DIRECTION
1	FWD	INPUT
2	BACK	INPUT
3	LEFT	INPUT
4	RIGHT	INPUT
5	+5V	———
6	TRG 1	INPUT/ OUTPUT
7	TRG2	OUTPUT
8	OUTPUT	OUTPUT
9	GND	———

Figure 3. Analogue input port assignments.

Figure 4. Pin out of the cassette I/O port.

PIN No.	SIGNAL NAME	DIRECTION
1	GND	———
2	GND	———
3	GND	———
4	CMTOUT	OUTPUT
5	CMTIN	INPUT
6	REMOTE+	OUTPUT
7	REMOTE−	OUTPUT
8	GND	———

PIN NAME	SPECIFICATION
1. VIDEO OUTPUT AND COMPOSITE VIDEO	DIN 5 PINS CONNECTOR OR RCA 2 PINS CONNECTOR
2. RF MODULATED SIGNAL	RCA 2 PINS CONNECTOR
CASSETTE	DIN 8 PINS CONNECTOR
I/O PORT	AMP 9 PINS CONNECTOR
PRINTER	UNPHENOL 14 PINS CONNECTOR
CARTRIDGE BUS	2.54 PACE, 50 PINS CONNECTOR
AUDIO	RCA 2 PINS CONNECTOR

Figure 5. The connectors used for the various ports form part of the MSX specification.

The 'switches' at addresses 82h and 83h are responsible for configuring the RS232 channel. 82h sets the baud rate with bits 0-3 setting the receive rate while bits 4-7 set the transmit rate. Setting the appropriate nibble (four bits) to a hex value in the range 0h to Ah will set the rate to one of the following rates – 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600 or 19200. The values Bh to Eh are unassigned while a value of Fh will disable the channel.

Location 83h takes care of the rest of the channel's configuration. Bit 0 is the CD line connected directly to the RS232 connector while the other 7-bits perform the following functions. Bit 1 selects auto line feed if set (1), bit 2 switches between full and half duplex (full if set). Bit 3 determines the XON/OFF control enabled if set and bit 4 determines word length; 7-bits if clear, 8 if set and bit 5 sets parity – even if set. Bit 6 enables parity if high and finally bit 7 sets the number of stop bits, 2 if set, 1 if clear.

Other areas of the I/O map will not concern the majority of users, the exception perhaps being the PP1 port. This is allocated as follows:

A8h R/W Port A
A9h R/W Port B
AAh R/W Port C
ABh Mode Register

Although the manual documents the I/O locations it warns that some manufacturers may not follow the spec. exactly (what price a standard?). In order to avoid any problems that this would cause, it suggests that all I/O is handled via operating system calls.

Which pin and connector review?

The diagrams of **Figures 1 to 5** show the pinouts of the various connectors of an MSX computer and need little in the way of explanation. In addition, the full specification of the VDP's modes is given in **Table 4.**

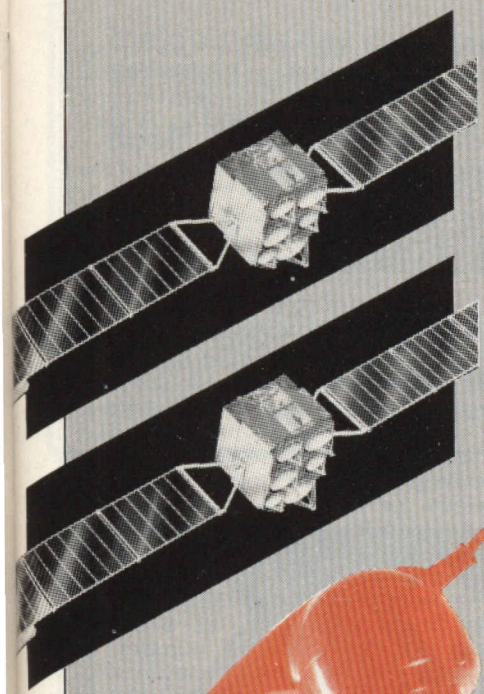
The end . . . for the moment

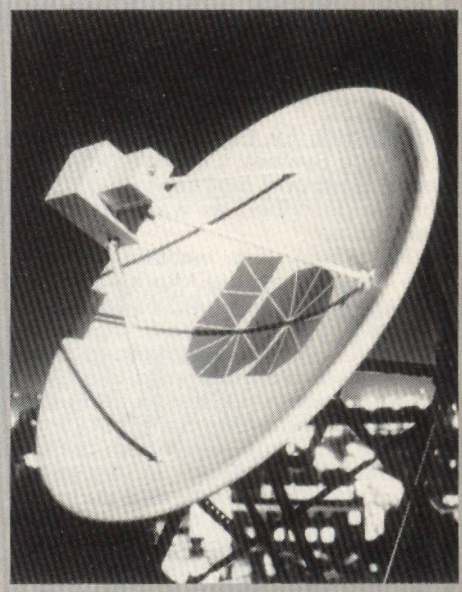
The formation of the MSX standard has taken account of the various demands that micro users put on their machines. The specification provides a flexible framework on which additional hardware can be constructed. The MSX standard goes beyond the definition of a mere games machine, providing a versatile system which should meet the needs of those interested in exploring the hardware and software potential of the machines.

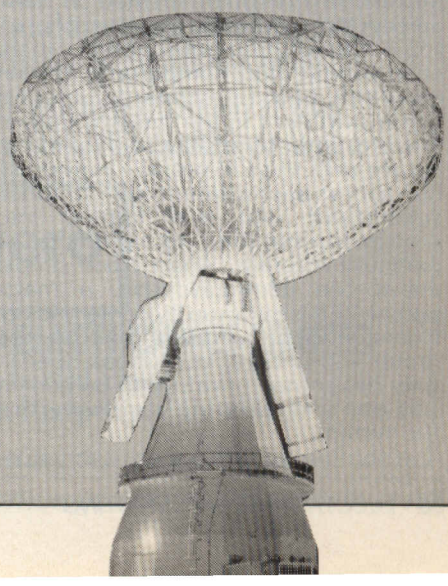
COMMUNICATIONS SPECIAL

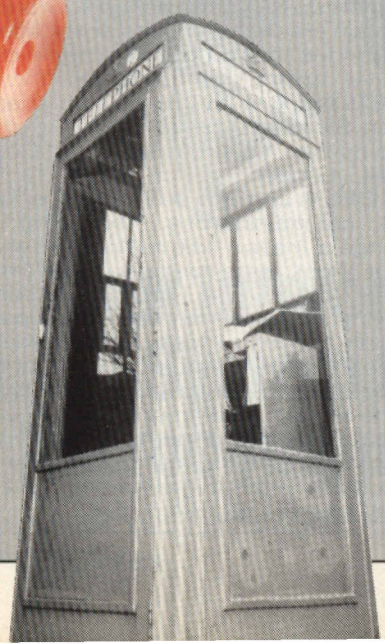
William Owen sets the scene for our communications special.

μ COMMUNICATIONS SETTING THE SCENE









Communications is now the leading edge of computer technology. Improvements in telecommunications have opened up new possibilities for linking computers together to access each others' software, data, and processing facilities; to send electronic mail, receive news, weather reports and information. The business world has already realised some of the possibilities, and now, as with every other field of computing, the technology has filtered down to the home micro user.

Few Spectrum or BBC micro owners realise that they have a potential mainframe terminal or electronic messaging system in their hands: the only extra equipment they need is a modem with suitable software. It is also possible, with a little extra hardware, to download weather pictures from satellites, or with suitable software, to emulate a full DEC communications terminal.

The possibilities which networking on the phone lines gives to the computer owner are immense. The obvious application of the new technology is access to information databases, not only in the UK but also, with the advent of packet switching and satellite links, in the USA – and for a price the home computer owner can afford.

Radio hams are using computers for a variety of uses, including morse transmission and reception, automatic station tuning, and radio telegraphy.

Databases and bulletin boards have proliferated in the UK since BT began and Prestel service; these are examined by Liz Gregory in the article on page 38, and all the information necessary to access them yourself is given. One area lagging behind – in terms of cost and frequency of usage, is electronic mail systems such as Telecom Gold. One day these will be available to the home user for transmission and processing of words. Katherine Custance has taken a look into the future of these systems and others which will soon be available, including cable and cellular radio.

Another, more exciting prospect is the networking of large groups of computer user. Once this could only be done on any scale by using large mainframes, but it is now possible by using a standard 16-bit personal computer with suitable storage space. William Owen has been researching one such system already in use in the United States, and talking to the people in the UK who are starting the work of building such facilities for home computer users.

Also included in this special supplement is a detailed description of the basic technical concepts involved in communication, together with a glossary of terms.

EIES TO THE FRONT

EIES or Electronic Information Exchange System, is a network on a grand scale. The system was established some ten years ago and today offers a vision of the future of communications.

EIES stands for Electronic Information Exchange System; it is not a database, nor an electronic mail service: EIES can only be described as a network on a grand scale.

The system was established in the mid-70s by Professor Murray Torroff of the New Jersey Institute of Technology. Professor Torroff originally worked on ARPANET – the supposedly secure (it's been broken into a few times) large scale packet switched network set up by the US military for the exchange of information by scientists working on defence projects. EIES is similar to ARPANET, but was set up primarily as a sociological study to examine the way in which individuals communicate via computers. As such the project has been a great success, and points a way forward to the future in a way which even interactive databases such as Prestel do not.

The system structure

The secret of EIES lies in its non-hierarchical structure. Most databases are output from a central souce. Input from readers is restricted to simple notice board which are given low priority in the pagination. In other words, these systems are little different from a newspaper with a classified advertising section. Electronic message services, on the other hand, are fully interactive but have no central databank or routing system – the emphasis is on privacy and confidentiality. As you will see, EIES takes its lead from neither system.

The nucleus of EIES is a Perkins Elmer mainframe computer located in New Jersey. The machine is capable of handling a total of 3000 subscribers – there are currently 1900, compared to 1200 in mid-1983. Each subscriber provides his or her own input to the system, firstly in the form of a short description of themselves and their interests. Individuals can access each other directly, or those subscribers who are formed into groups/associations to promote a particular interest. Apart from simple one-to-one communication, a major part of EIES is taken up by conferencing. Conferences are usually initiated by the group subscribers, but unless specifically designated private, can be accessed and addressed by any user within the system.

A non-hierarchical system is therefore one which allows and encourages input from the user rather than the system operators. Communication is possible between individual and individual, individual and group, and between group and group(s). The role of the system operator is restricted to ensuring ease of access between the subscribers. The operator acts as a guide – when requested – to take a subscriber to the particular part of the system, or subject, in which he or she is interested in.

Torroff is said to worry little about who uses the system; the more diverse the users the better in order to study different types of usage and behaviour. A couple of large US companies have bulk group subscriptions for their staff. Other groups vary from scientists to journalists to political organisations, and of course there are a good number of hackers who want to find out for themselves how the global village of the future will work. The conferences and conversations reflect the pre-occupations of the users, and will range from discussions on nuclear power, aspects of science and computing, politics and philosophy. Users can also arrange to co-author notebooks for presentation to conferences.

Packet switching is a cheap form of digital data transmission over telephone lines or via satellite. It is based on data compression techniques, enabling the inclusion of far more channels on one line than the wide band voice data links. A large amount of data from many different sources can therefore be sent over the lines to an equivalent number of destinations.

Messages are split up into small fixed sized pieces (the packets), each with a destination address contained within it. In most PSS networks each packet is also given a sequence number for reassembly of the message into a coherent order, even if the packets making up the message have been sent by different routes at different times.

Obviously including extra data about destination and sequence increases the cost and time involved in sending the message, but in the long run it is much cheaper: if a long message of several thousand characters is sent, while very little address information would be needed it would require a slow speed communications channel, and, if many such messages were being sent the network would become unstable.

Thanks to the packet switching system EIES is an international resource. There are subscribers in many European countries, including Britain, and in Asia and the Pacific.

Two British users are Len Stuart of the Association of Amateur Computer Clubs, and Sabine Kurjo, who has established a Peace Network also has pages on Micronet's Clubspot.

Logging on

Using the Packet Switched System is extremely cheap. All that is required is a computer (any), a modem, and the use of a number on the PSS network which can be obtained from registered users, for example Telecom Gold or Cable and Wireless, or directly from BT. A ten minute transfer of packet data from Britain to the US is measured in terms of pennies rather than pounds.

The subscription to EIES itself is rather more expensive, at $75 per month for an unlimited number of accesses. The high price is inevitable in what is essentially a non-commercial enterprise, but the advantage is that users are given an extremely good service in terms of guidance through the system, availability, and ease of communication.

Systems like EIES, if not EIES itself, will become much cheaper as time passes. Already EIES is (in terms of its technology) outdated: two New York systems analysts have designed a similar network to run on the IBM PC (with Winchesters) which will be able to handle up to 900 subscribers. Groups in Britain are discussing the possibilities of extended networking for home computer users, in particular through NETLINK, a study group established by Len Stuart and based at the Room 97 computer resource centre in County Hall, London.

EIES is perhaps a vision of the future of communications, extending the range of mass media not in terms of output from large organisations (newspapers, television, databases) but by giving individuals and small groups the ability to communicate their own ideas – telephones and letters could never offer this facility because they are essentially one-to-one communication. Large scale networking may go some way to providing such a means if allowed to blossom without interference. You can say anything you like on EIES.

MICROCOMPUTER DATABASES

The advent of low cost modems has enabled many people to join the micro communication revolution. Liz Gregory takes a look at the ever expanding range of databases accessible to anyone with a computer, a modem and a 'phone line.

The quality of facilities available to the user via micro/modem communication continues to increase with a corresponding fall in the price of necessary hardware (a modem). This is encouraging as micros can be put to a greater variety of communication uses without incurring huge expense.

Besides information that may be available through databases operating with a 1200/75 baud modem – the Prestel standard – an additional 300/300 baud facility will allow interactive communication between users via bulletin boards. These are computer 'noticeboards' which are springing up all over the country for enthusiasts who require information, need questions answered or simply wish to contact other users. They often have facilities for taking messages or orders if the board is a commercial concern, like Cashtel for example.

"Micros can be put to a wide variety of communication use without huge expense".

Along with the British boards listed below there are many international facilities now available to the user with the relevant modem and these include links with Scandanavia and the States particularly. Accessing overseas may prove difficult and users should check that they have appropriate facilities on their hardware, as the States operate a different protocol to most European countries. Britain and most of Europe use the CCITT standard whereas the US operates on the Bell standard.

Viewdata provides a more substantial information service than, for example, Teletext because it is interactive and operates via a central computer which is capable of both sending and receiving messages from subscribers. Britain's Viewdata service is Prestel which has around 44,000 subscribers who use the great variety of services available to micro owners. Not all databases operate under the Prestel system; for example Compunet, which is a new service available for Commodore owners, uses a private database called ADP which has 12 different computers in operation.

Prestel does offer many diverse activities ranging from teleshopping and

Name: London TBBS
Phone: 01-348 9400
Baud Rate: 300
Times: Mon-Fri 22.00 – 01.01
Weekends
14.00 – 01.00

Name: Manchester
Phone: 061-427 3711
Baud Rate: 300
Times: Sun-Thur 22.30 – 00.01
Fri 23.30 – 02.00
Sat 22.30 – 02.00

Name: Cashtel
Phone: 0702 552941
Baud Rate: 300
Times: 24 Hours

Name: Microweb
(Micro User magazine)
Phone: 061-456 4157
Baud Rate: 300
Times: 24 Hours

Name: Rewtel
Phone: 0227 232628
Baud Rate: 300
Times: 24 Hours

Name: Southern BB
(Chichester)
Phone: 0243-511077
Baud Rate: 300
Times: 24 Hours

Name: Stoke ITEC Remote CP/M
Phone: 0728-265078
Baud Rate: 300
Times: 24 Hours

Name: Surrey CBBS
Phone: 04862-25174
Baud Rate: 300
Times: 24 Hours

Name: South-West
Phone: 0626-890014
Baud Rate: 300 and 1200/75
Times: Ring and ask

Name: Forum 80
Phone: 01-902 2545
Baud Rate: 300
Times: Ring and ask

Name: West Midlands
(Stourport)
Phone: 0384 635336
Baud Rate: 300
Times: Daily 17.30 – 08.30

Name: BABBS (BASUG)

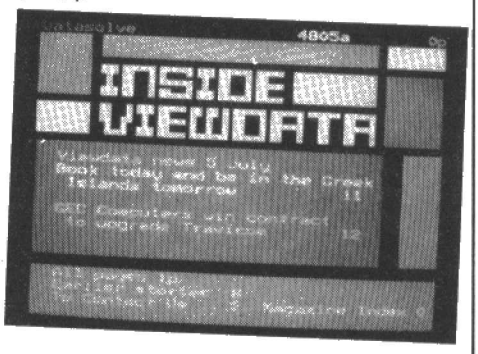
looking to improve the service and save users some money by introducing a national mailbox scheme whereby subscribers can gain access to their local Prestel computer instead of paying for longer distance phone calls. These local rate calls will save a lot of on-line charges.

One of the services on offer to users of Prestel is Micronet 800 which has brought around 10,000 subscribers onto the viewdata system. This is part of the general microcomputing section and Micronet subscribers can gain access to the services on offer within this particular part of Prestel without any additional charges. It provides all kinds of features including computer news, news stories and features, hints and tips on computing, and software which can be downloaded directly to the user's machine (usually for an extra charge), are keyed into a bishopgate terminal – a rapid on-line editing device which enables features to be sent to the mainframe and thus appear quickly on the screen's of subscribers. Other facilities include electronic mail, telesoftware and other areas for information and interaction between users.

"Micronet are expanding their scope of activity".

Micronet subscribers may send messages to one another either by mailbox, which is like a personalised letter; or via a response frame where a message is put onto the appropriate 'noticeboard' for other users to see. Thus, people wishing to know any computing tips would look at the pages set aside for this.

However, at the beginning of August, Micronet will introduce Chatline as a new method of communication between users. The service will allow for realtime conversations between subscribers who are on-line to those particular pages. Thus issues may be discussed in an electronic conference as different arguments arise. There will be 100-200 or so pages available during an experimental period and the service will be operable initially between 8-12 pm.



For games enthusiasts Micronet have introduced Starnet which is an interactive game capable of coping with up to 1,000 players at one time. The object is to capture as many stars as possible from your starting position, the co-ordinates of which you are given when you join. No financial aid is given and thus the more valuable stars must be the object of capture. Players may make alliances with other participants but there is a danger that they may be let down or even double-crossed. Moves are made on a daily basis which gives players plenty of time to make decisions. Designed by Mike Singleton, Starnet is already in operation and has several hundred players.

The majority of Micronet subscribers are currently BBC owners but this could be changed with the introduction in mid August of facilities for Commodore owners. Probably aimed directly at Compunet, Micronet are hoping to boost subscription with this move and interested parties will be able to purchase the relevant software of £49.95 or a Modem 1000 complete with software for £129.00. Given the cheapness and quality of some of the available modems, users may wish to buy the software and a device which also operates under 300/300 baud as well as the 1200/75 rate required for Prestel. Thus they can use their hardware for accessing bulletin boards as well as viewdata. There should be around 80 programs available for Commodore users and about half of these will be free.

Further developments which Micronet are planning for the future are an educational facility called 'Headstart' which will be a weekly program aimed at a 5-12 age group with the editorial aimed in sections at parents and children. Users will also be able to download relevant software. An executive Micronet service is planned and this will have standard business letters available so that details may be simply keyed in before hard copies are produced. A directory of CP/M software will also be included with this service which will appear later on this year.

the downloading of telesoftware to information provision for a number of different groups including farmers and businessmen. They are currently

Phone:	0742 667 983
Baud Rate:	300
Times:	24 Hours
Name:	Birmingham – North
Phone:	0827 288810
Baud Rate:	300
Times:	24 Hours
Name:	Blandford Board
Phone:	0258 54494
Baud Rate:	300
Times:	24 Hours
Name:	Chiltern CBBS
Phone:	07073 28723
Baud Rate:	300
Times:	Evenings only 18.30 – 22.30
Name:	City Bulletin Board
Phone:	01 606 4194
Baud Rate:	1200/75 – Wednesdays 300 – all other times
Times:	24 Hours
Name:	Computer Answers (Magazine)
Phone:	01 631 3076
Baud Rate:	1200/75 – temp. out of order 300
Times:	24 Hours
Name:	Cumbria CBBS
Phone:	069 92314
Baud Rate:	300 & 1200/75
Times:	18.00 – 22.00
Name:	Distel
Phone:	01-679 1888
Baud Rate:	300
Times:	24 Hours
Name:	Distel
Phone:	01-679 1888
Baud Rate:	300
Times:	24 Hours
Name:	Hull Forum 80
Phone:	0482 859169
Baud Rate:	300
Times:	Mon-Fri 17.30 – 23.30 Sat & Sun 12.00 – 23.30 7 days 00.00 – 08.00 a week (Bell 103)
Name:	Mailbox 80
Phone:	051 428 8924
Baud Rate:	300
Times:	24 Hours
Name:	London CBBS MG-NET
Phone:	01 399 2136
Baud Rate:	300
Times:	Sundays 17.00 – 22.00

COMMUNICATIONS

1984 and beyond

Katherine Custance takes a look at the future of communications between computers.

PHOTO TOM EVANS MODEL MIKE POPE

When Alexander Graham Bell invented the telephone, he was dismissed as a crank by the leading American telegraph company. The people could not understand why anyone would want to speak to someone else over a line when they could send a telegraph and have a hard copy at the end.

As there was no established base of telephone owners there would have been no-one to talk to anyway! This logic has slowed down many developments in communications. Before you can start a service you need at least two customers and to make money you need many more. Also if people are happy with the old technology, why force them to accept the new?

Consequently, in the early days of micros, manufacturers didn't bother investigating communications. The beauty of the micro was that you could use it on its own. It had its own power and so you didn't need to use it as a terminal to a larger machine. This meant that the development of consumer communications equipment lagged behind the development of consumer electronics.

But the liberalisation of British Telecom, which allows other companies to sell their equipment direct, and the enormous growth of microcomputer owners in Britain has changed the attitude of manufacturers. Even the cheapest micro now has the option of a communications interface and the price of modems – the devices that change the digital computer signal into an analogue signal that can travel down a telephone line – has plummeted.

A couple of years ago, few people could have predicted that some modems would now cost under £50, and things are now moving so fast in the communications industry that companies are no longer planning more than one or two years ahead.

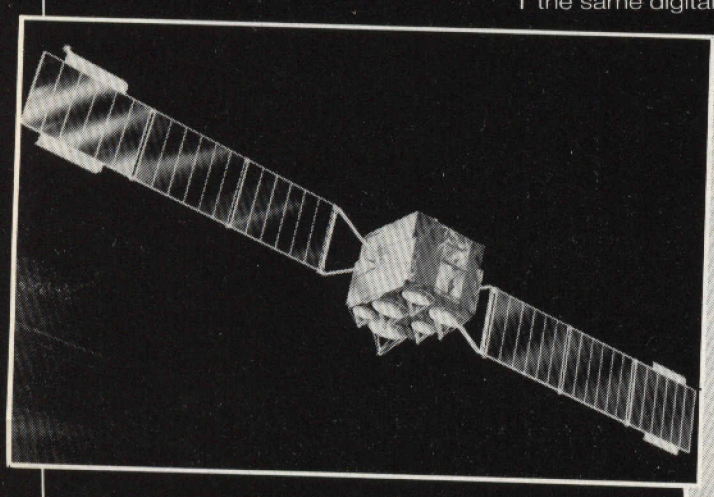
But although hard and fast predictions are impossible, there are some definite trends that can give us a taste of microcommunications in the 1990s.

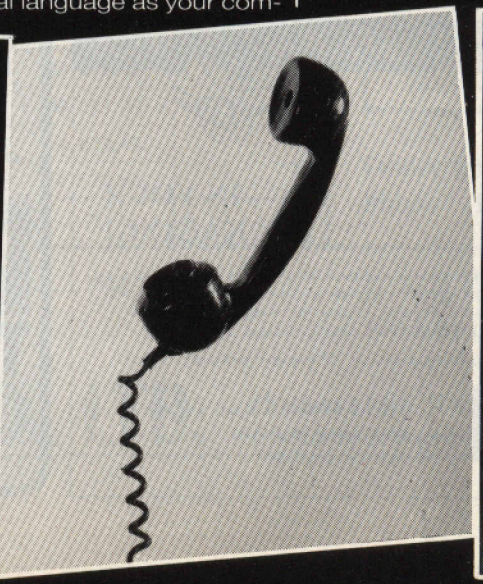
On the hardware side each micro will have a built-in modem chip, which will cost pennies rather than tens of pounds to produce. The standard interfaces will become even more 'standard' and instead of buying or writing communications software, all the information your computer needs will be burnt into the memory. There will also be more flexibility so you could send information at many different speeds and in different formats.

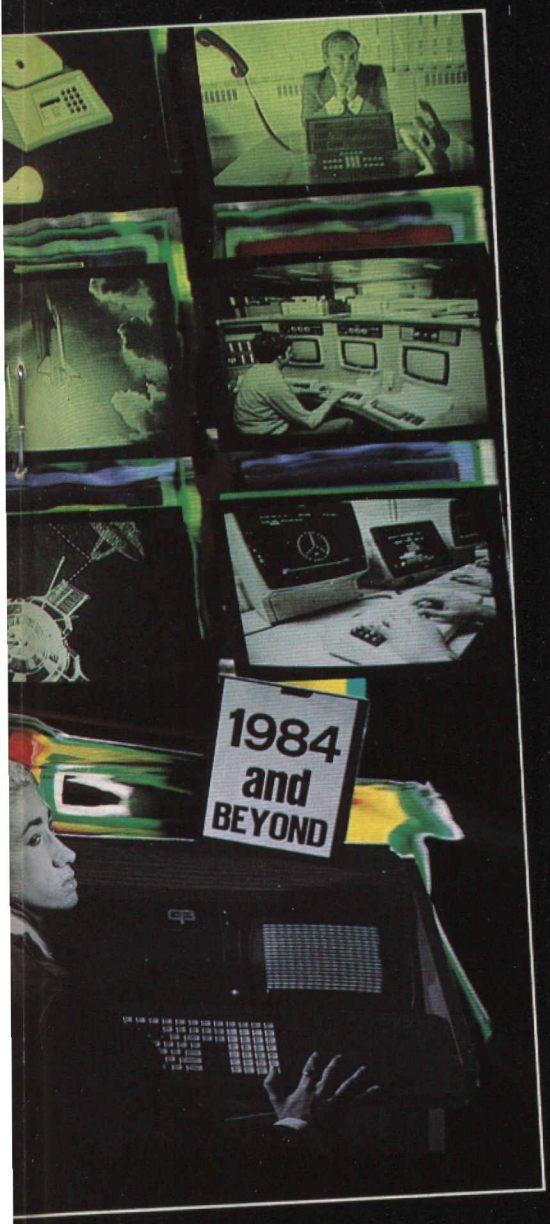
Optical fibre cable will replace metal cable to connect pieces of equipment because it will be cheaper as mass production gets underway – sand, which is the main ingredient in making glass, is dirt cheap! The advantage of sending data as light pulses down a fibre is that it is not distorted by electrical interference or radio waves, so that the quality of the signal is better.

On a larger scale, as Britain gets wired up with digital telephone exchanges the whole quality of the service will improve. Eventually, by the end of the century, there will be no need for today's modems, because the network will be talking the same digital language as your computer.

In the next decade your home could have another communications network – cable TV. Like the telephone network this will be two way in most cases, so that you can send and receive information. But having the hardware is of little use unless there are the services there. Many of you will have already







experimented with sending and receiving messages, either directly to another micro or through one of the electronic mail services (such as Telecom Gold), Prestel, Micronet or a bulletin board.

These fledgling services are already popular, but can be quite expensive for a home micro user. Some of the electronic mail services can also be complicated for the uninitiated. Ideally, sending an electronic message should be cheaper than posting a letter or making a phone call. Some electronic mail companies already claim they can do this, but unless you're a frequent user all the other charges make each message quite expensive.

In the next year we will see the cost of electronic mail slashed as the companies realise the enormous potential in the home micro market. The number of commands needed to access these services will also drop so that they become easier to use.

More and more service providers will make use of the packet switched network (PSS), British Telecom's high speed data network. This alows them to offer micro users local call access. This means you can dial a local number to get onto the PSS network, rather than dialling directly into a computer in London or New York, so saving money on your telephone bill.

Some mail services offer links to the telex network, so, in effect, you can send a telex from your home. In the future, mail services will have more of these links or 'gateways', so that you can contact virtually any device from a printer to a mainframe, by a number of different networks both inside Britain and abroad. This is all obviously useful, but all it really does is improve on what went before – it helps you contact someone faster and more cheaply than you previously could.

What will really revolutionise the home micro is access to large databases, not only to download software, but as a source of information. There are many large databases stored on mainframes throughout the world with both general and very specialised information on them. Traditionally the owners of these databases make their money by charging low volume business users very high rates. This is changing.

An American company, Dialog Information Services, has just launched a new database service in the UK, called Knowledge Index. This is a scaled down version of the original Dialog database. The information stored by the two services amounts to 100 billion characters covering an enormous range of subjects. The initial cost is £25, which includes two hours of access. After that it costs only $24 per hour (paid at the current exchange rate) plus the cost of a local call, even though the computer is in America. So a five minute interrogation would cost not much more than a pound. It uses a word search method which gives you a lot of information in five minutes.

Over the next few years, these systems will mushroom with endless applications both for work and leisure. Access to database and remote access to company computers will undoubtedly accelerate the trend towards 'homeworking', where people work at home for part or all of the week.

Apart from the astronimcal savings on overheads for businesses this will make an impact on the whole of society – the way people structure their home lives, where they choose to live and the leisure and transport industries. Working from home in this way does not need to be expensive, all you need is a cheap micro and an acoustic coupler.

'each micro will have a built-in modem chip'

The other most exciting trend in microcommunications is portability. At the moment your micro has to be near a power point and a telephone socket to send messages; in the coming years it will be able to communicate nationally and internationally from a car, a train or the middle of a field.

People have used car telephones for a number of years, but the sophistication of the new 'cellular' radio network, which will come into operation next year, will allow you to send and receive data through a radio network. Initially this will be at fairly slow speeds and users would be advised to stay in one place, but once the technology problems are sorted out these restrictions will disappear.

Racal is now involved in a project with Acorn and others to install micros, screens and telephones into cars so the driver can receive up-to-date traffic information and other database information. The idea is to eventually have a system which is cheaper than the cost of today's car telephones which will recognise spoken requests for information and be able to speak back. Maybe the science fiction dream of talking to someone on the other side of the planet using your wrist watch is not too far from reality.

Allophones and Orthography*

Or how to make the most of your speech synthesiser. By John De Rivaz.

*Correct or conventional spelling; spelling with reference to its correctness for those of you who were wondering.

With the arrival of the SPO256AL2, allophone chip, speech synthesis from the hardware point of view has become a trivial matter. However, programming allophones is not particularly easy, and the following software has been developed to meet that need. The illogicality of English orthography is such that it would be a hard job to make a microcomputer accept text and convert it to speech. Apart from being difficult to write, the program would also be extremely long.

This program takes the input string letter by letter and gives the operator a choice of allophones to use. Although it is perfectly possible to input a whole sentence (or indeed the works of Shakespeare!) as the input string, it is better to have one word at a time. Each word can then be tested and added to another string which is the spoken sentence. The spoken string is represented as a series of ASCII characters.

Readers who have a speech peripheral other than the *Electronics and Computing Monthly* design (which nevertheless uses the same chip) should check that it also uses port 159, or otherwise alter lines 740 and 750 to suit the port used.

Correction to circuit

In order to make the *E&CM* circuit function properly and reliably, the two unused gates in the 74LS00 should be used to buffer the oscillator from the speech circuit. If they are not so used, the speech circuit may not be clocked properly, as the voltage from the oscillator may not swing sufficiently near to zero.

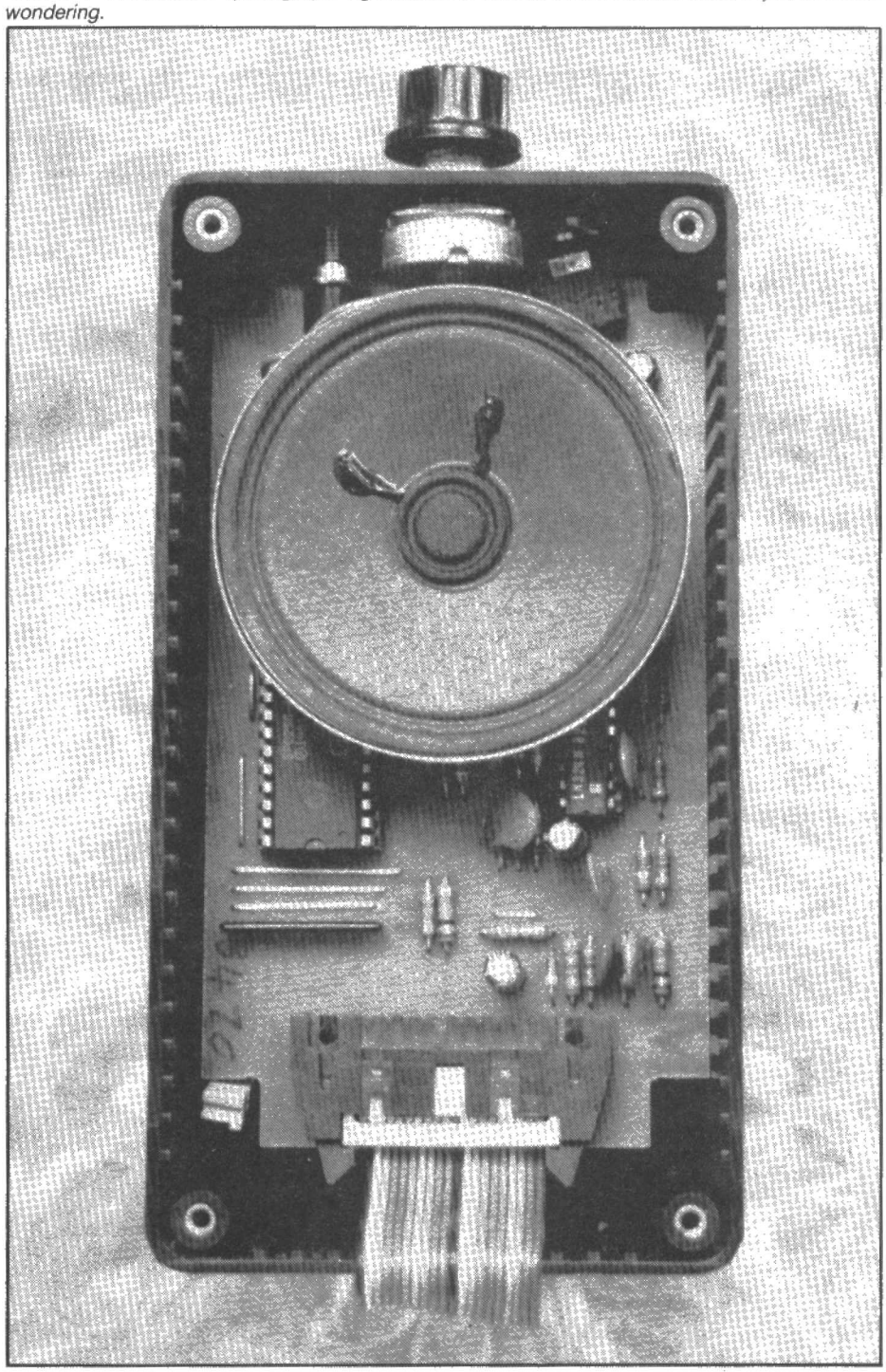
Obtaining the program

You may wish first to refer to the Spectrum manual on how to obtain different colour papers in a string. (Press caps shift and symbol shift simultaneously followed by the colour of paper required.) When entering the DATA lines, some of the characters are underlined. Make these appear on yellow paper by pressing CS+SS6 before the underlined letters, and CS+SS7 afterwards.

Using the program

When RUN for the first time, you are presented with a blank screen with the string input message "What do you want me to say". Enter the first word of your message, for example "Halt", then press ENTER.

You will be presented with the word "halt" stripped of capital letters followed by a reverse block **h** and a request to press one of eight numbers. Opposite each number is a word with one or more letters on yellow paper. Assuming that you want the **h** to sound like the **h** in hose, then press 2. You then get the word printed again, this time with the **a** on yellow paper, and another sequence of numbers and words. If you decide that the **a** in "halt" is like the "au" in "aught", then press 1. Then do the same with **l** and **t**. Then the computer will say "halt" over and over again until you press ENTER.



Watford's new synthesiser (above) provides a ready made collection of words.

If you didn't like the way it said the word and want to try again, then press **r** and repeat the process.

Assuming that you approved, then press **a** to add it to the sentence you are composing. You'll get a beep to show this has been done. If you wish, you may then press **i** to hear the sentence so far. (Obviously in this case it will just say "halt".)

If you want to add a pause, then press **r** to enter the next word, and enter a couple of periods "..". You will, of course, be greeted by silence. But press ENTER to get the menu, and add these to the sentence.

The next word is "you". Press **r** to start it, and when you get the prompt, enter the word. The first letter **y** is straightforward, press **5** for the **y** in "yes". Remember, the **o**

is not sounded, so when asked for this, press **s** for skip. You might think that the **ou** is like the **u** in "computer", therefore press 1. However, when you try this, is doesn't sound right. Therefore, repeat the word and then press 7 for **u** in "sue". Once satisfied, press **a** to add it to the sentence.

Assuming you don't want a pause here, the next word is entered; "will" has no peculiarities, except the final **i** is skipped. "be" is the next word, and then, finally, "exterminated". This demonstrates that **x** is automatically entered, (as is **qu**). You now have a string:

?;Q1'=C&ROQ&c7&'+M[1X40/81+9$

which you can LPRINT, save on cassette or just copy off the screen to add into your own program based on a popular TV series! There is no copyright paronoia about lines 720 to 770. Readers of *E&CM* may use them to their heart's content in their own program or for publication!

Program notes

The listing starts with the main data areas. The data is made up in blocks, one for each letter of the alphabet. The first item is this letter, followed by example words using it. These are followed by a slash (/) and their respective allophones. The reason it is presented in this way is that the program can

LISTING 1

```
 10 REM Speech software development aid (c) RTL 1984. Available on cassette from
 20 REM  RTL, Westowan, Porthtowan, Truro, Cornwall TR4 8AX for £2 post free.
 30 REM
 40 REM sent for publication 25 May 1984
 50 LET s$=CHR$ 39
 60 DATA "a","aught","fat","great","hair","farm","clear","pinta","/","23","26","20","47","59","60","51"
 70 REM       --      -     --      --      --      ---       -
 80 DATA "b","rib","big","/","28","63"
 90 REM         -   -
100 DATA "c","computer","sky","crane","church","/","42","41","8","50"
110 REM       -        -   -       --
120 DATA "d","could","do","/","21","33"
130 REM         -   -
140 DATA "e","bend","see","great","label","computer","herd","clear","/","7","19","20","62","51","52","60"
150 REM       -      --      --      --        --      --      ---
160 DATA "f","fire","/","40"
170 REM       -
180 DATA "g","beige","guest","go","wig","bang","/","38","36","61","34","44"
190 REM       --   --      -     --    --
200 DATA "h","he","hoe","whig","church","they","bathe","thin","shirt","/","27","57","48","50","18","54","29","37"
210 REM       -    -     --     --       --      ---    --     --
220 DATA "i","fitting","sky","hair","bird","wig","/","12","6","47","52","48"
230 REM       -       -     ---    --     --
240 DATA "j","judge","/","10"
250 REM       - --
260 DATA "k","computer","sky","crane","/","42","41","8"
270 REM       -        -     --
280 DATA "l","label","luck","/","62","45"
290 REM         --    -
300 DATA "m","milk","/","16"
310 REM       -
320 DATA "n","earn","no","bang","/","11","56","44"
330 REM       -    -     --
340 DATA "o","aught","cot","cook","toy","to","food","out","snow","store","/","23","24","30","5","22","31","32","53","58"
350 REM       --      -     --     --    -    --     --      --      ---
360 DATA "p","pub","/","9"
370 REM       -
380 DATA "qu","q1","2","42","46"
390 REM qu special
400 DATA "r","hair","computer","bird","store","farm","clear","read","brain","/","47","51","52","58","59","60","14","39"
410 REM       ---      --       --      ---     --      ---     -       -
420 DATA "s","sat","shirt","/","55","37"
430 REM       -     --
440 DATA "t","they","bathe","thin","its","to","/","18","54","29","17","13"
450 REM       -      ---     --     --    -
460 DATA "u","computer","succeed","aught","out","curd","guest","sue","/","49","15","23","32","52","36","22"
470 REM         -        -       --      --     --      --       --
480 DATA "v","even","/","35"
490 REM       -
500 DATA "w","whig","wool","/","48","46"
510 REM       --     -
520 DATA "x","x1","41","55"
530 REM x special
540 DATA "y","sky","toy","bay","typical","yes","/","6","5","20","12","25"
550 REM       -     --     --      -         -
560 DATA "z","zoo","/","43"
570 REM       -
580 DATA ",,",",","space",".",":",";","/","2","4","3"
590 DATA "28","63","21","33","36","61","34","10","50","\","17","13","42","41","8","9","29","40","55","27","57","37","48","/"
600 GO TO 800
610 REM **** NUMBER ONE TO NINE IN ALLOPHONES
620 DATA 43,12,12,39,53,0: REM ZERO
630 DATA 46,23,11,0: REM ONE
640 DATA 13,31,0: REM TWO
650 DATA 40,14,19,0: REM THREE
660 DATA 40,58,0: REM FOUR
670 DATA 40,6,40,0: REM FIVE
```

easily be modified if some extra allophones are added in later. It may be found that some additional modifications can be thought of. (An example is "sue" under **u**.) Refer to page 19 of January 1984 *E&CM*.

The data at line 590 may be used to insert stops if required. The data is allophones that require stops, and the inverse slash (/) is used to differentiate between fields.

Next follows the data for the numerals, as given in the original *E&CM* article, and after this comes the subroutine to, say, a string composed of the allophones plus 36. The reason why 36 is added is so that these strings are visible when printed, and can therefore be used in magazine published listings. The characters " and # were avoided, as they may upset DATA statements and some printers.

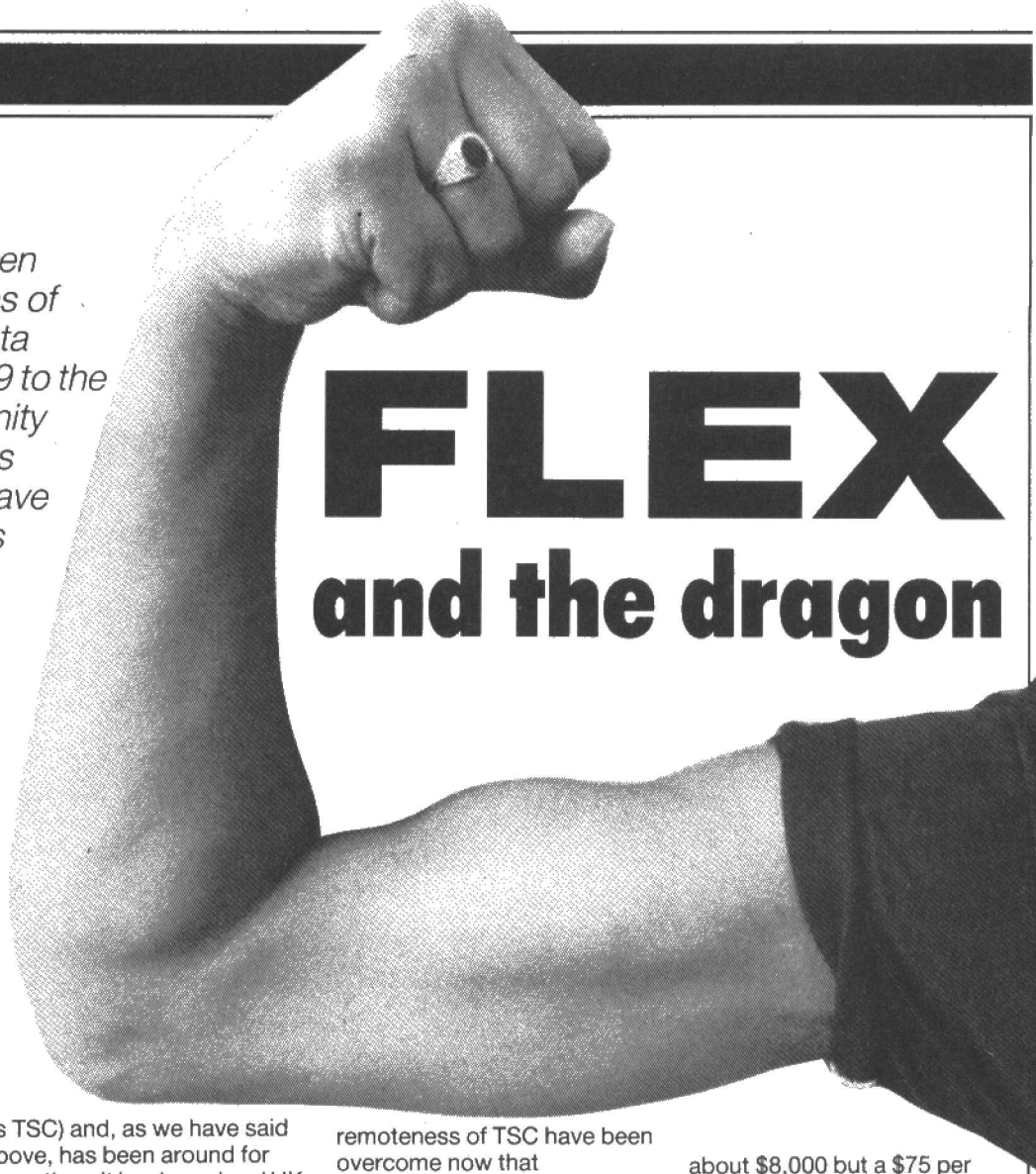
Line 800 starts the program itself. The string inputted is taken letter by letter, and each letter is used to select data as previously described. X and G get special treatment and are made up of two allophones as shown in their respective DATA statements. These can be changed if required.

At line 1030, a check is made to see if a stop is required before actually adding the allophone to the string a$. In 1100, double voiceless fricatives are added.

The final menu is dealt with in lines 1120 onwards. ■

```
680 DATA 55,12,2,41,55,0: REM SIX
690 DATA 55,7,35,7,11,0: REM SEVEN
700 DATA 20,2,13,0: REM EIGHT
710 DATA 56,6,11,0: REM NINE
720 REM subroutine to say phrase a$
730 FOR n=1 TO LEN a$
740 IF IN 159>127 THEN  GO TO 740
750 OUT 159,(CODE a$(n)-36)
760 IF a$(n)<>" " THEN  NEXT n
770 RETURN

780 REM main loop
790 REM =========
800 INPUT "What do you want me to say? ";b$
810 LET a$="": FOR j=1 TO LEN b$:
820 IF b$(j)>="A" AND b$(j)<="Z" THEN  LET b$(j)=CHR$ (CODE b$(j)+32)
830 IF b$(j)<"0" OR b$(j)>"9" THEN  GO TO 870
840 RESTORE 620+10*VAL b$(j)
850 READ c: IF c=0 THEN  GO TO 1110
860 LET a$=a$+CHR$ (c+36): GO TO 850
870 IF b$(j)>="a" AND b$(j)<="z" THEN  RESTORE (60+20*(CODE b$(j)-97)): GO TO 990
880 RESTORE 580
890 CLS : PRINT b$( TO j-1); PAPER 6;b$(j);: PRINT b$(j+1 TO ): PRINT
900 LET c=0: READ c$: PRINT  INVERSE 1;c$
910 READ c$: IF c$="/" THEN  GO TO 960
920 IF c$="x1" THEN  FOR g=1 TO 2: READ c$: LET a$=a$+CHR$ (36+VAL c$): NEXT g: GO TO 1110
930 IF c$="g1" THEN  FOR g=1 TO 3: READ c$: LET a$=a$+CHR$ (36+VAL c$): NEXT g: LET j=j+1: GO TO 1110
940 PRINT "Press   for"
950 LET c=c+1: PRINT c;",  ";c$: GO TO 910
960 PRINT "s,      skip"
970 LET d$=INKEY$: IF d$="" THEN  GO TO 970
980 IF d$="s" OR d$="S" THEN  GO TO 1110
990 IF d$<"1" OR d$>STR$ c THEN  GO TO 970
1000 FOR d=1 TO VAL d$: READ c$: NEXT d
1010 REM find whether stop required
1020 REM ==========================
1030 RESTORE 590: LET shift=0
1040 READ e$: IF e$=c$ THEN  LET a$=a$+CHR$ (37+shift): GO TO 1070
1050 IF e$="\" THEN  LET shift=1: GO TO 1040
1060  REM add allophone to string
1070 LET a$=a$+CHR$ (VAL c$+36)
1080 REM double voiceless fricatives
1090 REM ============================
1100 IF j>1 THEN  IF b$(j-1)=" " AND (c$="29" OR c$="40" OR c$="55") THEN  LET a$=a$+c$
1110 NEXT j: LET a$=a$+CHR$ 36
1120 CLS : PRINT "This is the word string for yourprogram:"'a$'"This is the sentence string  foryour program:"'s$'"Press any
key to continue."
1130 GO SUB 730: PAUSE 20+10*RND: IF INKEY$="" THEN  GO TO 1130
1140 OUT 159,0
1150 PRINT "Press e to end                    h to hear word again        i to hear sentence          s to SAVE word
          a to add word to sentence    d to SAVE sentence          c to LPRINT sentence       r to produce another
 word "
1160 LET c$=INKEY$: IF c$="" THEN  GO TO 1160
1170 IF c$="h" THEN  GO TO 1120
1180 IF c$="c" THEN  LPRINT s$: LPRINT : GO TO 1160
1190 IF c$="s" THEN  DIM z$(LEN a$): LET z$=a$: SAVE "word" DATA z$()
1200 IF c$="r" THEN  LET a$="": GO TO 800
1210 IF c$="i" THEN  LET e$=a$: LET a$=s$: GO SUB 730: LET a$=e$: GO TO 1160
1220 IF c$<>"a" THEN  GO TO 1270
1230 LET s$=s$( TO LEN s$-1)+CHR$ 39+a$: BEEP .1,.1: LET a$=""
1240 IF LEN s$>2 THEN  IF s$(1)=CHR$ 39 THEN  LET s$=s$(2 TO ): GO TO 1240
1250 IF LEN s$=0 THEN  LET s$=CHR$ 39
1260 GO TO 1120
1270 IF c$="d" THEN  DIM z$(LEN s$): LET z$=s$: SAVE "sentence" DATA z$()
1280 IF c$<>"e" THEN  BEEP .2,.2: GO TO 1160
```



FLEX and the dragon

FLEX and OS9 have been fighting for the affections of 6809 users. Dragon Data recently introduced OS9 to the home computing fraternity and now, as Lee Francis reports Compusense have configured FLEX for this computer.

When it comes to operating systems designed for the 6009 processor the MPU has what can only be described as an embarrassment of riches. Dragon Data managed to launch the OS9 operating system for their computer shortly before their current financial problems and this excellent 6809 OS and its Dragon implementation is covered elsewhere in this issue of *E&CM*.

OS9 is however just one of two operating systems designed for the 6809. The other contender in the field is TSC's FLEX which has just been released in a version tailored to the Dragon. The battle between OS9 and FLEX has been going on for a number of years and each OS can point to users that consider its features to be better than the other. Unbiased observers though often concede that FLEX has the edge in a number of areas.

This article does not set out to be a users guide to FLEX nor indeed will it concern itself with a comparison between FLEX and OS9 – these matters will be the subject of later features; instead this brief piece will attempt to give a little background information on the FLEX OS and to examine the way in which it has come to the UK market.

Compusense and FLEX

FLEX was developed by Technical Systems Consultants Inc USA (hereafter referred to as TSC) and, as we have said above, has been around for some time. It has has a loyal UK following but, as is so often the case, in this context the word loyal goes hand in hand with small. The problem is a common one in that TSC in the States inevitably find it hard to offer the sort of support that a package such as FLEX demands. This has put many people off adapting FLEX for their machines. The other reason that FLEX has not had a wider UK reader base is that, until recently, there has not been a great deal of low cost hardware on which to run the OS. The *E&CM* Hi-res computer was based on the 6809 processor and FLEX was configured for this machine. Unfortunately constructing the computer was a job for somebody with a great deal of hardware knowledge and this prevented many people from tackling the project. The Dragon computer and indeed Tandy's Co-Co are the only popular UK machines to have adopted the 6809 processor and it comes as no surprise that FLEX is now available for these machines.

The difficulties of the remoteness of TSC have been overcome now that Compusense have been appointed UK distributors of the package but perhaps distributor is the wrong word as it implies a fairly passive role for Compusense.

The form of the licence that Compusense have been granted is unique and allows the Company to distribute FLEX, its editor and assembler on any designated hardware. This overcomes one of the major factors when determining a manufacturer choice of OS, namely cost.

Terms of endearment

The agreement allows Compusense to offer FLEX to an OEM for a small initial fee plus a unit price that is based on volume of order. This is in contrast to the previous arrangements when a typical fee would be of the order of $18,000 for the end user and this would be restricted to a single item of hardware. An alternative, though still expensive opinion was the part licence arrangement when the initial fee would be smaller, about $8,000 but a $75 per copy royalty was charged. These figures did not, by the way, include any configuration work on the part of TSC.

In contrast Compusense are prepared to undertake configuration work, although must charge for this, and can offer FLEX implementations for as little as £50. This is the sort of price a home user is charged for Dragon Data's OS9 and as the question of support for this OS must be called into question at present, FLEX is an attractive proposition for many Dragon users.

Compatible software

One of the attractions of the OS9 package was the number of utility packages that Dragon managed to produce for running under the OS: word processor, spreadsheet, database etc. The situation with FLEX is, if anything, slightly better and there are a wealth of packages written for FLEX. Once again Compusense are handling the distribution of these

The future

There can be no doubt that the 6809 and either FLEX or OS9 have major attractions for the home user. The fate of the Dragon computer is still in the balance but with the arrival of 6809 second processors for the BBC micro, many people will have the chance to set up a system capable of running either OS.

We shall be looking at some of these second processors and at Flex in next month's *E&CM*.

FLEX applications

While some computer users will kit themselves out with either OS9 or FLEX in order that they may gain experience of using a computer system under control of a 'real' operating system, many more will see the OS as merely a means to an end – the end being to run applications software.

Before describing some of the many applications packages available it would be as well to look at the sort of price you can expect to pay for typical items of software.

So, returning to OS9 for a moment, one of the major attractions of Dragon Data's launch of this OS, together with a suite of application software, was that they were able to maket it at such low cost. The complete range of items each managed to break the sub-£100 barrier, in some cases by a comfortable margin. The company managed to achieve this by obtaining an 'outright licence' from the copyright holders Microware. This process meant that Dragon purchased considerable numbers of the various packages and paid for them 'up front' – this according to reports involved a sum of money in the region of $200,000. It is ironic that this action probably contributed to the demise of Dragon although it is still difficult for others to compete on cost of software.

The situation is much the same in the States where an outright licence was obtained in respect of Tandy's (or Radio Shack's) colour computer which is known as Co Co in the US. The Co Co is virtually identical to the Dragon in both its hardware and software and so the parallels with the UK market are interesting. The effect of Co Co's outright licence is to dramatically reduce the price of FLEX or OS9 software for the machine – it is typically half the price of standard packages.

Neither the Dragon nor Co Co is capable of running standard software as this demands a single density disk format, whereas the home machines have adopted a double density format. Standard software will not BOOT although, with a little ingenuity it is possible to write a single density driver – but that's another story.

There is also a hurdle that prevents the running of some software configured for the Co Co on the Dragon. This is because the Co Co does not have an equivalent of the Dragon's 51 column mode which causes difficulties when running some packages.

The best source of information about both FLEX and OS9 software is a magazine called Soft News the first issue of which recently appeared in the US. This magazine is in effect a revamped version of the Frank Hogg Laboratories catalogue. It goes beyond the concept of a catalogue and not only includes details of the vast range of packages distributed by the company but also contains a number of interesting articles about the two OS's.

Compusense will be distributing the catalogue in this country and an A4 stamped addressed envelope will secure a copy of Soft News.

£'s to $'s

The dollar prices quoted in the catalogue convert rather painfully into pounds (hardly surprising in view of the current exchange rate and transportation costs) and in addition to that there is the dreaded VAT at 15%.

Compusense will be happy to import any item in FHL's magazine although the company foresee that £100 will present a cost barrier and see most sales having a value below this figure. For this sort of money it will be possible to get FLEX versions of Stylograph, database packages and spreadsheet software.

The battle between FLEX and OS9 for the affections of 6809 users has been going on for a number of years now. OS9 is the younger of the two operating systems and does have the advantage of being 'UNIX like'. As UNIX is likely to become increasingly common in the future OS9 makes an ideal vehicle for 8-bit micro users to gain some familiarity with a look-alike. Meanwhile the old guard see OS9 as a young upstart and point out that in many areas FLEX has advantages over OS9.

In the end it is down to a matter of personal preference. However, in this country the battle could well be won by OS9 if only because it costs so much less than FLEX because of that outright licence of Dragon Data's.

Compusense's address is 286D Green Lanes, London N16. 01-882 0681.

```
 NAME     TYPE   SIZE     P          .CMD     1
                          PROT       .CMD     1
 FLEX     .SYS     33     RENAME     .CMD     1
 ERRORS   .SYS      9     S          .CMD     1
 PRINT    .SYS      1     SAVE       .CMD     2
 SERIAL   .SYS      1     SAVE       .LOW     2
 APPEND   .CMD      3     SDC        .CMD     3
 ASMB     .CMD     48     STEP       .CMD     1
 ASN      .CMD      1     TTYSET     .CMD     2
 BACKUP   .CMD      4     VERIFY     .CMD     1
 BAUD     .CMD      1     VERSION    .CMD     1
 BUILD    .CMD      1     XOUT       .CMD     2
 CAT      .CMD      3     STARTUP    .TXT     1
 COPY     .CMD      5     WELCOME    .TXT     4
 CS       .CMD      1     DEMO1      .TXT     2
 DATE     .CMD      2     CS         .ITA     2
 DELETE   .CMD      2     CS         .UK      2
 DRIVES   .CMD      1     CS         .FRA     2
 EDIT     .CMD     28     CS         .GER     2
 EXEC     .CMD      1     CS         .DEN     2
 H        .CMD      1     CS         .SWE     2
 I        .CMD      1     CS         .USA     2
 JUMP     .CMD      1     CS         .SPA     2
 LINK     .CMD      1     CSTEST     .TXT    22
 LIST     .CMD      3     SERSYS     .TXT     4
 NEWDISK  .CMD     10
 O        .CMD      2     SECTORS LEFT = 471
```

Directory of FLEX master disc showing the wide range of facilities provided by the OS. We shall be returning for a closer look at FLEX in use in a future issue of E&CM.

RS423 and the BBC

In this extract from a McGraw-Hill book, Colin Opie investigates the BBC micro's RS232 channel. The book, Interfacing the BBC micro, is available from leading bookshops at £8.95. It contains a wide range of useful information on the BBC.

This port is implemented through the joint efforts of a 6850 ACIA and a customised ULA chip. It is only necessary to know about the 6850 ACIA in order to be able to adjust the use of the serial interface, all other aspects being well covered by operating system calls.

The 6850 ACIA has only two access addresses, as shown in **Figure 1.** The first address (SHEILA &08) is for the status and control registers and the second address is for the data I/O registers. Notice that it is not possible to write to the status register or read the control register. **Figure 2** shows a simplified diagram of the internal registers and how they are interconnected.

Control register

The operation of the 6850 ACIA within the BBC computer is such that a little caution is required over how, and which bits of, the control register may be altered. As far as this text is concerned the discussions will be limited to the alteration of serial data formats only.

Figure 3 shows that control register bits 2 to 4 determine the format of the serial data. As three bits are used, eight options are open to us. The BBC computer defaults to option 5 (ie 8 bits, no parity, and 1 stop bit). Altering the serial data format is extremely simple and achieved by using the 'FX' call:

*FX 156,n,227

where 'n' is the operating system format value shown in **Figure 3.** Notice that this value is the same as four times the binary pattern value for the appropriate bits in the control register. It would not be at all common for this data format to be changed in any one application and therefore the 'FX' form of operating system call is quite adequate. Always remember that this call must be used in order to ensure that the other bits in the control register do not become adversely affected.

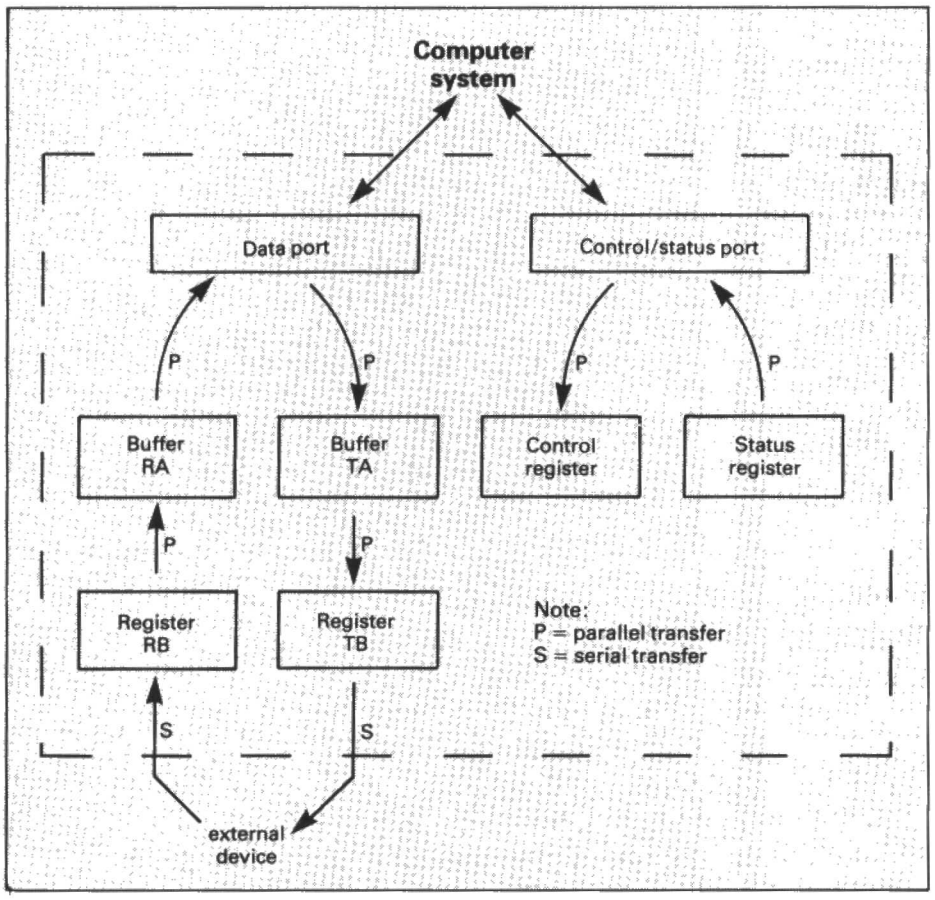


Figure 2. 6850 ACIA register layout.

Status register

Provided you keep to the limitation defined above and only use the RS423 interface for simple I/O transfers, there are only three bits within the status register which are pertinent. (Other bits, in both the control and status registers, are mainly concerned with interrupts and error detection).

The 6850 ACIA status register is a little like the interrupt flag register of the 6522 VIA. In other words, it contains flags which indicate some activity, but these flags do not necessarily generate interrupts – they may be either monitored or used to generate IRQ signals. **Figure 4** shows the three bits which are of use to us. Note that, as with most 'flag' registers, bit 7 is used to indicate that it was this device which generated an IRQ signal.

Programming the RS423 port

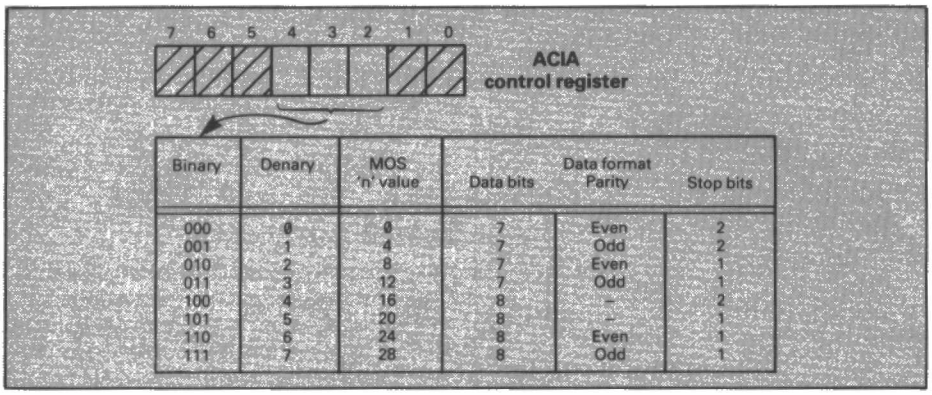
Sending and receiving data through the RS423 port is simply a matter of taking account of the flags in the status register and accessing the input and output data buffers.

When inputting serial data it is necessary to watch out for bit 0 of the status register (RDRF). This bit will go 'high' when a byte is transferred from RB to RA. If the input data buffer RA is not read before the input logic has successfully collected a second byte, then the original one will be lost, because it will be overwritten in the buffer.

When writing data, two conditions need to be catered for. Firstly, the CTS bit in the status register (bit 3) must be 'low', as this indicates that the external device is willing to accept data (ie is not busy). A data byte should never be sent to the data output buffer TA if the CTS flag is 'high'. Secondly, and assuming that CTS is found to be 'low', the TDRE flag (bit 1) must be checked. When the internal logic transfers a byte from the output buffer TA to the serial output register logic TB, this TDRE flag goes 'high' – signifying that a subsequent byte may now be loaded into the output buffer. Ignoring either the CTS bit or the TDRE bit will cause bytes of data to be lost from the output stream.

Read	Write	Address (hex)	SHEILA offset (hex)
Status register	Control register	FE08	08
Data input buffer	Data output buffer	FE09	09

Figure 1. 6850 ACIA internal registers.



Binary	Denary	MOS 'n' value	Data format: Data bits	Data format: Parity	Data format: Stop bits
000	0	0	7	Even	2
001	1	4	7	Odd	2
010	2	8	7	Even	1
011	3	12	7	Odd	1
100	4	16	8	–	2
101	5	20	8	–	1
110	6	24	8	Even	1
111	7	28	8	Odd	1

Figure 3. Data format control.

Implementing a user-print routine

Selecting a printer type, on the BBC computer, is a simple matter of using the '*FX5,n' command. If '*FX5,3' is issued then a user-supplied routine is called, through the UPTV vector at address &222, whenever data is to be sent to a 'printer' (or to/from some other suitable device, ie, it does not have to be a printer).

User-print routines have to obey certain rules and respond to a given set of access types, called *reasons*. When setting up such a routine it is vital that the vector at &222 is set to point to the appropriate code before the '*FX5,3' command is given, in order that the software can be accessed during the actual initialisation procedure.

User-print routine operation

First and foremost the routine must take account of the use of Ctrl-B and Ctrl-C for the switching on and off of the printer output stream. Once this is taken care of, the method of transferring data will depend upon the interface being used and upon whether interrupts or monitoring operations are being performed.

For monitoring operations the following sequence of events should be catered for:

```
BEGIN
  IF NOT (printer-busy)
    THEN IF (buffer-empty)
           THEN set-dormant
           ELSE BEGIN
                  take-data;
                  send-data;
                  set-active
                END;
END.
```

Taking bytes of data from the printer output buffer is simply a matter of using an OSBYTE call with A=&91 (145). The buffer number (ie 3) must also be in the X register. On exit from the OSBYTE call the Carry flag (C) will be set (ie C=1) if the buffer was empty. If the buffer did have something in it then the Carry flag will be clear and register Y will contain the character retrieved. Sending the data out will depend purely upon which interface is being used. If you

LISTING 1

```
1000 REM SINT - Serial Interface Routine
1010 REM C.N.OPIE   Copyright(c) 1983 McGraw-Hill
1020 REM
1030 PROCset_interface
1040
1050 REM Main Program
1060 REM ------------
1070
1080 END
1090
1100 DEFPROCset_interface
1110 DIM code% 100
1120 FOR pass%=0 TO 3 STEP 3
1130 P%=code%
1140 [OPT pass%
1150 .ssend  PHP       \Save flags
1160         CMP #2    \Control-B issued?
1170         BEQ pen   \yes-set prt enable.
1180         BCC psnd  \No-but useful!!
1190         CMP #3    \Control-C issued?
1200         BEQ pen   \yes-set prt disable.
1210         BNE sext  \Not useful!!
1220 \
1230 .pen    STA pflag \Set/Reset prt enable
1240 .sext   PLP       \regain flags
1250         RTS       \and return.
1260 \
1270 .psnd   PHA       \Save registers
1280         TXA
1290         PHA
1300         TYA
1310         PHA
1320         LDA #&96  \Check printer busy
1330         LDX #8
1340         JSR &FFF4
1350         TYA
1360         AND #8    \CTS bit low?
1370         BNE psx1  \No-so exit!
1380         LDA pflag \Printer o/p enabled?
1390         CMP #2
1400         SEC
1410         BNE psx   \No-so exit!
1420         LDA #145  \Check printer buffer
1430         LDX #3
1440         JSR &FFF4
1450         BCS psx   \Empty-so exit!
1460         TYA       \Save character
1470         PHA
1480 .outw   LDA #&96  \Ok - write it out
1490         LDX #8
1500         JSR &FFF4
1510         TYA
1520         AND #2
1530         BEQ outw  \Wait for 6850 ready
1540         PLA       \Retrieve character
1550         TAY
1560 \
1570 \The char (in Y) is now ready to be sent. If
1580 \the char is to be modified - do it now!
1590 \
1600         LDA #&97  \Send it out (from Y)
1610         LDX #9
1620         JSR &FFF4
1630 \
1640 \The character has now been sent. If the char
1650 \sent requires other jobs to be done, eg.
1660 \null chars after a CR,LF - do it now!
1670 \
1680 .psx1   CLC       \Leave routine active
1690 .psx    PLA       \Restore registers
1700         TAY
1710         PLA
1720         TAX
1730         PLA
1740         PLP       \and flags.
1750         RTS
1760 \----------
1770 .pflag  NOP       \**Flag Store**
1780 \----------
1790 ]
1800 NEXT pass%
1810 ?&222=(ssend MOD 256):?&223=(ssend DIV 256)
1820 ?pflag=3:REM printer o/p inactive!
1830 *FX5,3
1840 *FX8,7
1850 *FX6
1860 ENDPROC
```

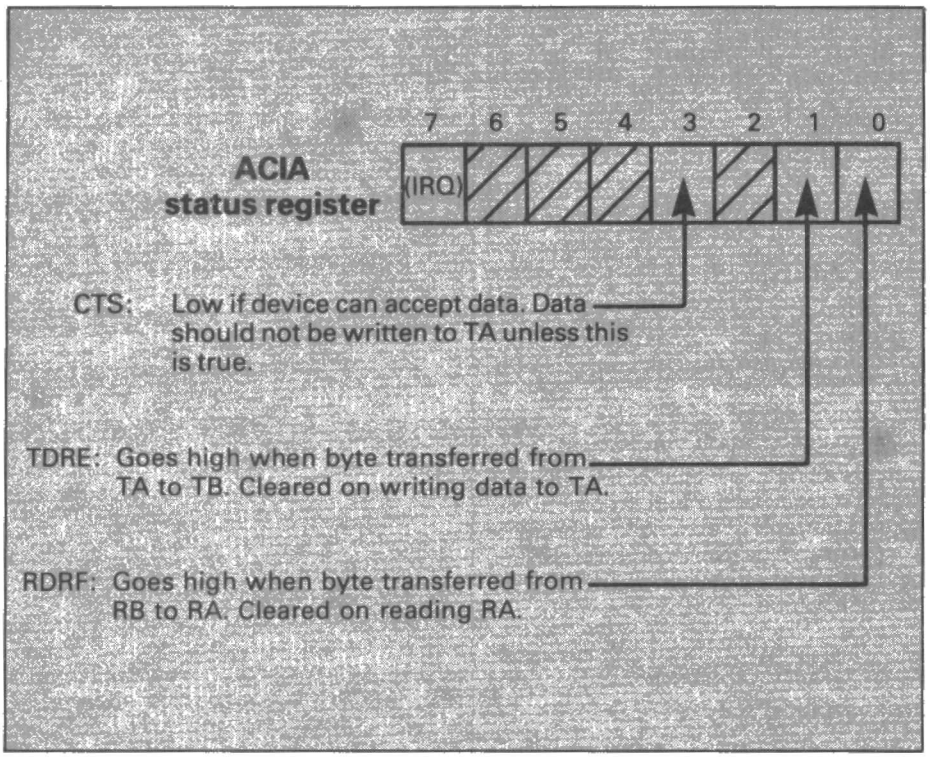


Figure 4. 6850 ACIA status registers.

are using interrupts then the interrupting device should be checked for, within your interrupt service routine, in the normal way. Making a routine 'dormant' can also be achieved by using an OSBYTE call with A=&7B (123). No parameters are required.

An example

The two most available and obvious interfaces to use are the already existent parallel printer interface and the RS423 port. With this in mind the example described below shows the use of the serial interface through user-print routines.

Serial port

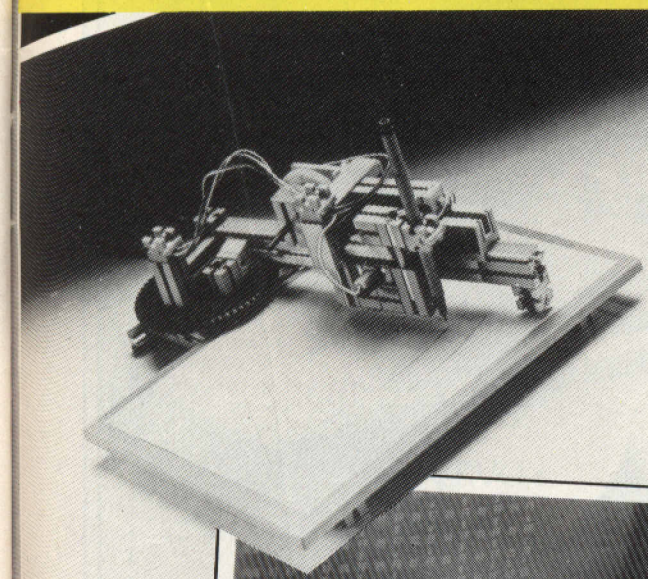
The program (called SINT) is shown in **Figure 5** and is entirely self-contained. Although it does not actually do anything in itself, it does show up some important points and will provide a convenient building block for your own routines.

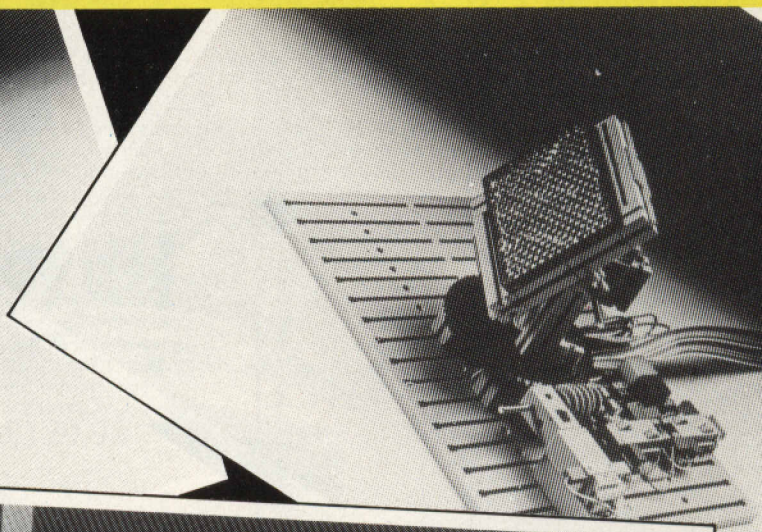
The main purpose of producing the program was to enable certain application programs to access different kinds of serial printers. As it stands in **Listing 1** it does not do anything apart from setting up a completely standard RS423 driver routine running at 9600 baud. However, there are two places near the end of the Assembler procedure which allow for both modification of data sent and subsequent operations to be performed if the data sent should require it. The definition of the task for this routine is such that reason code 5 is ignored. Notice that the vector at &222 is not set up until the code actually exists and that the user-print routine is not selected until the vector is set. It is significant that 'FX' calls may still be used to select the baud rate, and so on for the interface. ■

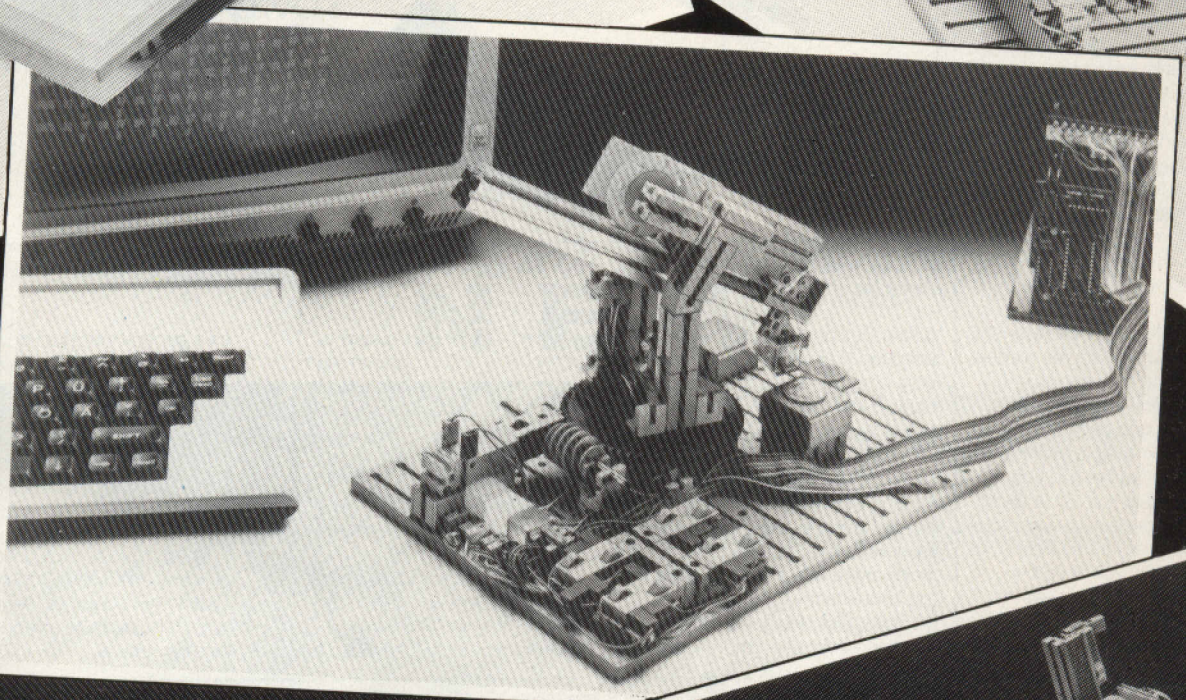
your ROBOT

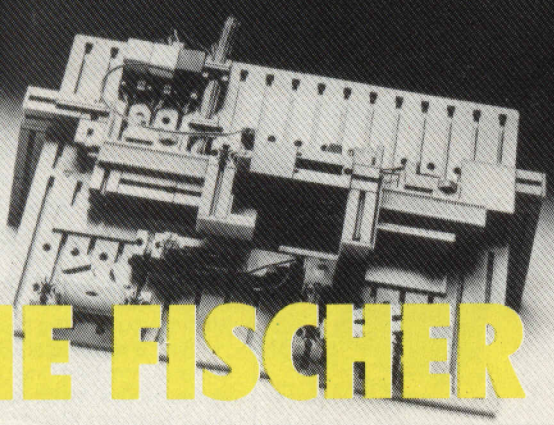
An EMAP Publication

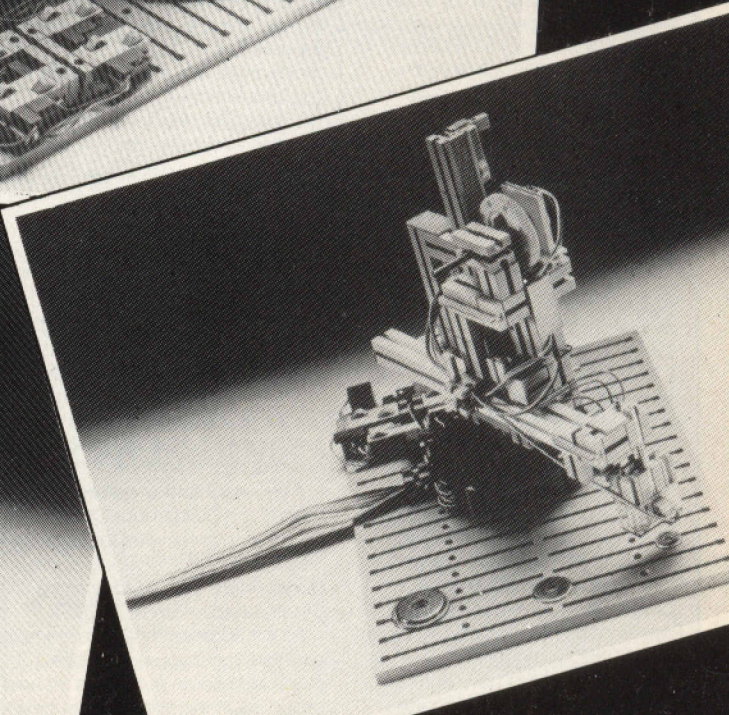
BRITAIN'S FIRST ROBOTICS MAGAZINE SEPTEMBER 1984











THE FISCHER TECHNIK TECHNIQUE

LATEST NEWS FROM THE WORLD OF ROBOTICS

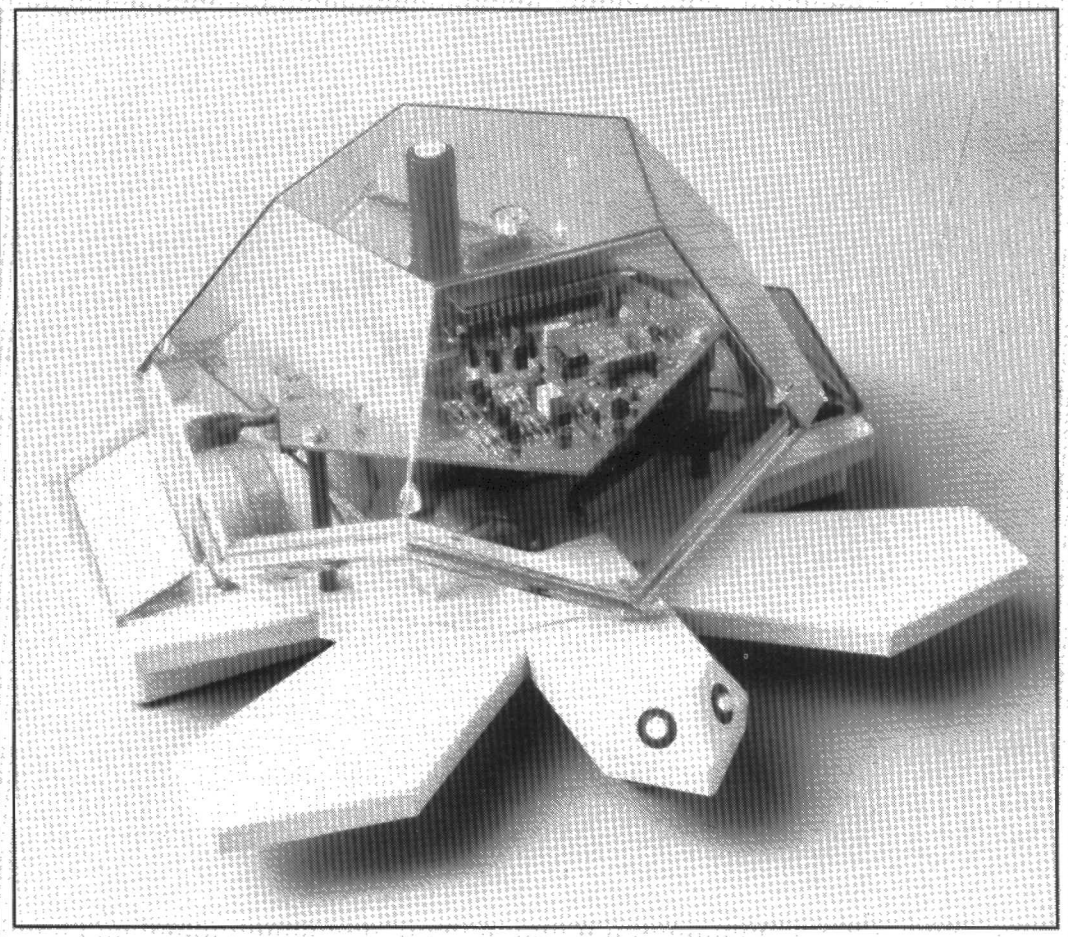
ARA NEWS

The Amateur Robot Association has announced that they are to take an interest in the Fischertechnik robot building kit which, as we report elsewhere in this issue, is now available in the UK after a successful launch onto the German market.

The Association which has its office at 26 Mill Hill, Weston Colville, Cambridge CB1 5NY, publishes a bi-monthly newsletter which will among other things use as a forum for experimenters with Fischertechnik computer controlled construction projects. The ARA which charges £4.50 for Associate Members and £12 for an annual subscription to the bulletin is requesting its readers to give details of their efforts, quandries and requests.

One interesting aspect of the ARA's activity is to provide connections with the American amateur robot organisations. The Robot Society of America (RSA) and Robotic Experimenters American League (REAL) both have a number of branches mainly on the West Coast. The various organisations are already talking to each other and among the thirty or so branches there are almost two hundred Americans who are building robots of many levels of sophistication. It would seem that amateur roboticists badly need a forum and lines of communications such as the ARA represents for coping with the detail of what is available on the amateur robot market.

That market for robot components and kits is not as well served as here as it is in America. The shops and retail outlets which specialise in robotic product and development are beginning to multiply over there. One of the best, Nu Tech in Los Angeles has begun a small chain of these shops. There is of course more disposable income in the US and the American hobbyist will often buy £2000 worth of robot whereas his British counterpart could not afford to do so.



The production version of the Valiant turtle. The turtle is supplied with a copy of Penup, a magazine introducing the principles of the LOGO language.

Valiant Design have officially launched their remote controlled turtle onto the home and education market. The turtle now has a smoothly moulded shell as opposed to the prototype previewed in the July issue of **Your Robot** and is capable of being interfaced with most home micros. This moulded shell shape appears to make the turtle look more sturdy in the wake of classroom boisterousness.

The turtle is designed with the aim of making education more fun as children can be introduced to computers and programming in particular with relative ease. The teaching of the language LOGO is naturally very important as it shows the fundamental approach to learning the rudiments of programming by moving an attractive futuristic device around and about.

The principles of LOGO are laid out in a glossy and well-produced magazine which comes as part of the Valiant package. PENUP is designed as a general introduction to the device and attempts to explain the usefulness of the language by suggesting ways in which to use the turtle. Children can be taught to program it to draw, move around a specific area like a race track, demolish building blocks and even to play games with it.

The company are confident of its success and with 4,500 orders from 15 different countries, the turtle looks as if it is going to be a very successful training vehicle.

CALLING ALL SCHOOLS

Building a robot is becoming increasingly fashionable. The easiest way to build your first robot is obviously with the aid of a kit. A number of kits are coming on to the market to show beginners the way to develoop an understanding of the theory and practice of computerised control of mechanical devices. The prices of these devices are reasonably modest so that many hobbyists and almost any school can afford their own robot. As a part of this movement probably the world's biggest robot building competition is going on at this moment. British Petroleum are running the BP Build a Robot 1985 Competition.

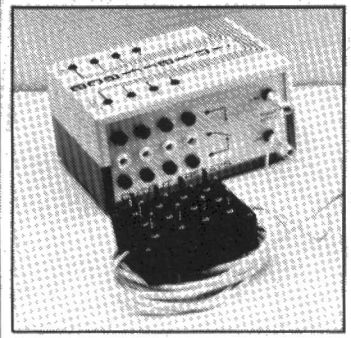
So far 40 schools have entered and another sixty have submitted their names and are hoping to enter the contest. Let's hope that they will fill out the rather thin numbers in The Midlands and Northern Ireland. This trend is a little surprising as both of these areas have a good record in robotic developments in higher education establishments. We will be monitoring the progress of the contest and individual school efforts over the next few months.

If you want further details of the rules of the Buildarobot Competition and how to enter write to *Freepost Licence No. 4482, Francis Parker, Room 0408, BP Oil Ltd, Victoria Street, London SW1E 5NJ.*

As the address implies no stamp is necessary and you will receive a poster and some guidelines on how to approach the task of robot design and construction.

BATTLESHIP INTERFACE

Pilot One Ltd. have recently completed development of an I/O interface for the BBC micro/Electron. The unit, named 'Interface' in a logical, if unimaginative fashion, is built like the proverbial battleship.



Pilot One's Interface.

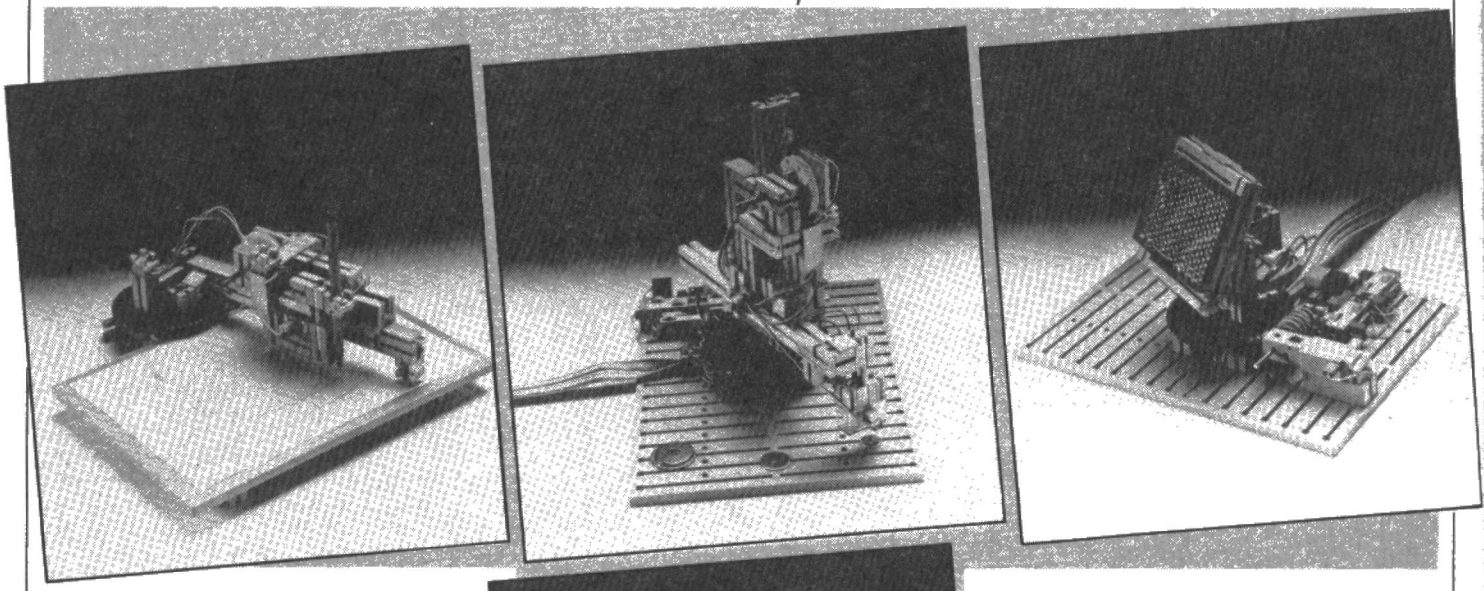
There is little chance that even the grossest misuse would harm either the interface or the micro to which it is connected.

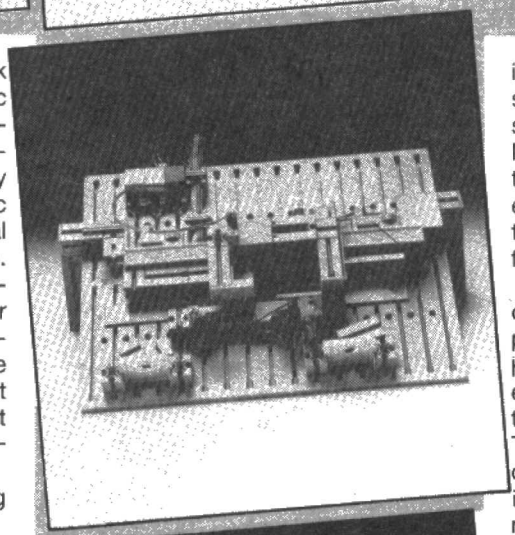
Interface is supplied with a suite of software that enables programs to be quickly written and tested. This software makes use of machine code routines yet it seems no knowledge of m/c programming on the part of the user. This emphasises Pilot One's application orientated philosophy; they want people to **do** things with their interface and not become bogged down in heavy and lengthy programming sessions.

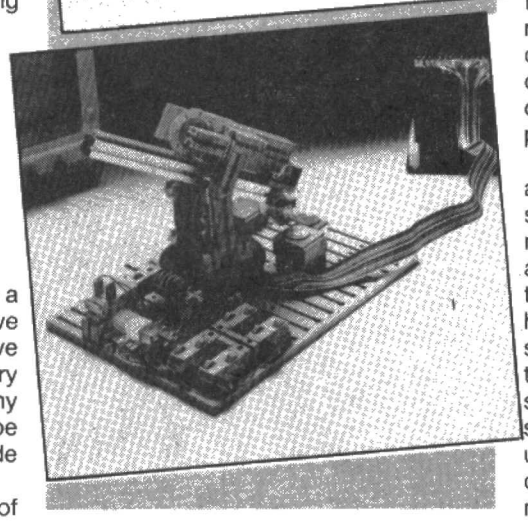
Interface is priced at £60 and is available from Pilot One Ltd at Victoria House, 46 St. Augustine's Road, Bedford MK40 2ND. Telephone 0234-781938.

FISCHERTECHNIK ROBOT KITS

The Fischertechnik robot builder kit has been on sale in Germany for some time now. The kit is about to make its UK debut and this should be of interest to anyone experimenting with low cost robotic systems. Peter Matthews assess the kit's potential.







The flexible nature of the Fischertechnik building system allows a variety of robotic devices to be constructed from a surprisingly small number of basic components. As our pictures show, the company in conjunction with Micro Robotic Systems, have developed six initial systems and there are more in the pipeline. Fischertechnik have exploited the versatility of the existing components of their technical sets and in addition have introduced some new components to aid in the development of robotic devices. Not least of these is a potentiometer housing that allows positional feedback to be incorporated as required.

The designs to date include the following systems:

A telescopic robot arm with magnetic gripper
A graphic tablet
A sorting machine
A lifting robot
A plotter
A solar tracking device

In addition a turtle type device and a lathe are in the course of design. We have never previously seen such a cost effective method for teaching simple control theory and robotic concepts. The kit will make any of the above devices so that one can be made up and then broken down and made into another.

An interface which enables the control of a Fischertechnik robot or plotter etc, with the BBC, Spectrum, Commodore 64, Vic 20, Oric or Apple is also available. The supporting software that accompanies the interface drives the small machines with some ingenuity. This adds a new dimension to the whole of the range of kits. Manuals are at present being prepared and these will contain experiments with the electronics, geometry and other aspects of the physics and maths of simple robot configurations.

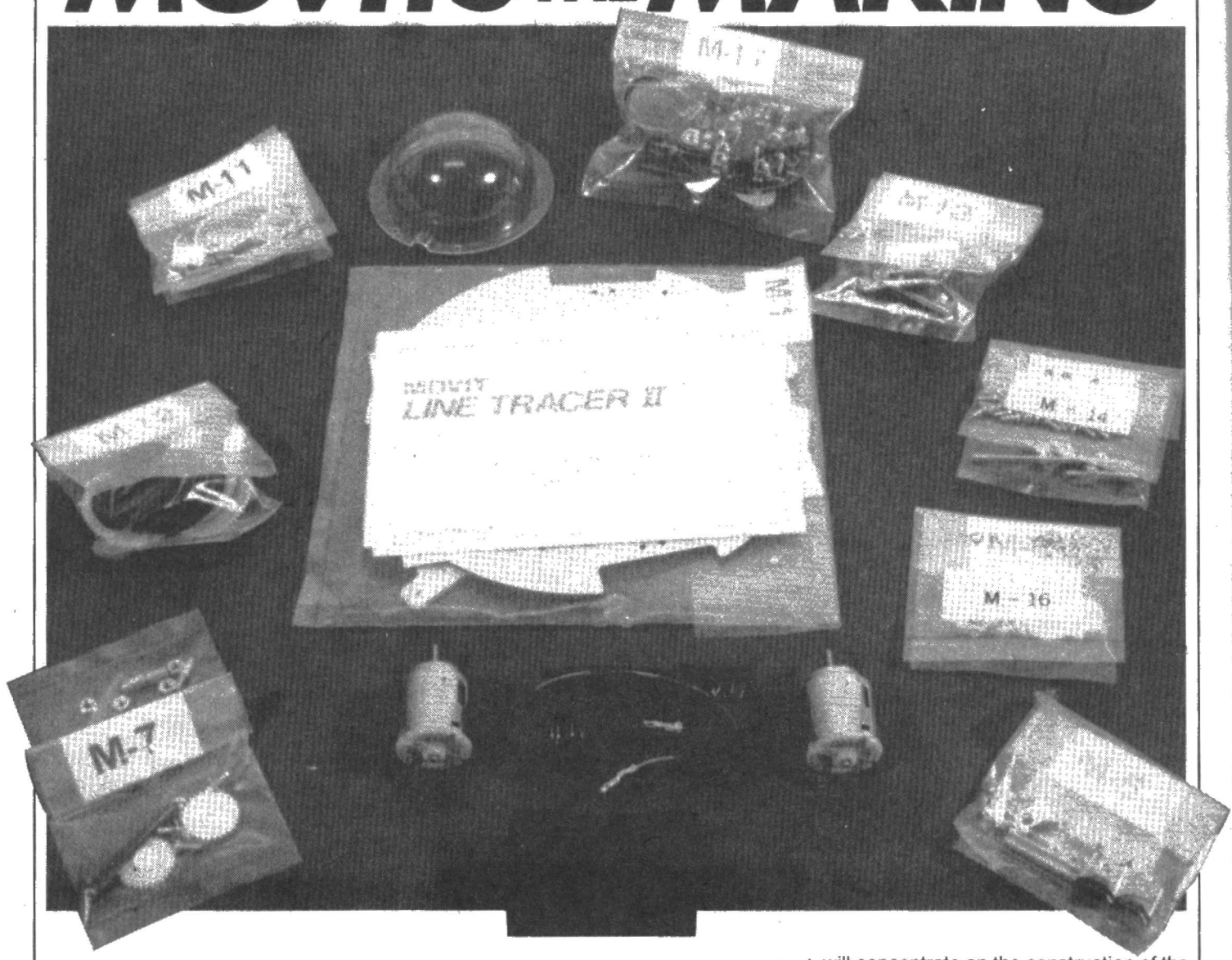
The five robots that can be constructed from the Fischertechnick robot builders kit.

Over the next few months we expect to describe and review Fischertechnick computer controller models individually. We have not had one long enough to test exhaustively yet but to date our favourite is the plotter that you see in the photograph. The kit contains all the materials and details for constructing the device including motors, microswitches, potentiometers, relays etc, including some details of the calculations and geometry for the device if you are mathematically inclined. It can be used, however, to simply draw pictures and patterns.

The plotter, as you can see, rotates its arm from the back of the board which swings over the paper. The pen holder is moved backwards and forwards by a rack and pinion device. This gives an X,Y position anywhere on the paper. The pen holder can be raised and lowered by a solenoid in the penholder. The Fischertechnick kit is operated by pressing switches and the photograph shows the simple patterns that have been drawn using this control. A simple interface gives computer control which makes it much more sophisticated.

This whole concept of a do-it-yourself computer control system for building blocks is likely to be extremely popular with the robot hobbyist.

MOVITS IN THE MAKING



Ken Alexander has put together the complete Movit range and reports on his experiences.

The Movit series of robots from Prism could well do for robotics what the MK14 did for micro computing. The family provide an ideal low cost introduction to the concepts of mobile robotic systems.

While nobody would claim that the Movits have the potential of products like Economatic's Buggy or Powertran's Hebot, at prices that range from just under £10 to about £35, it's not surprising that some sacrifices have been made in the specification of the Movits. The major drawback with the machines is that they have all been conceived as stand alone systems and consequently there is no provision for computer control of any Movits actions. Another failing is that there is no in-built positional feedback, a factor that rules out any turtle-like applications for the robots.

Before experimenting with Movits there is the small question of having to put together the kit of parts. All the Movits arrive as a collection of nuts, bolts, plastic and electronics, all packaged in a number of small bags. Each of these bags is numbered making the job of identifying the various components straightforward despite the vast array of components. However, before a single part is touched, it is recommended that the comprehensive instruction sheet is studied from start to finish. Failure to heed this advice can lead to a lot of additional work at later stages of construction – this is the voice of experience speaking!

LINE TRACING

We built up all of the robots in the family but will concentrate on the construction of the Line Tracer for the greater part of this article. Construction commences with the two drive motors, the first step being to fit a small pinion gear to the motors' shafts. The instructions suggest the use of a hammer to help the gear on its way but this rather drastic step was not found to be necessary; merely pushing the part home proved to be quite adequate. Having mounted the motor in its cradle, the next

"There is no provision for computer control of Movit's actions".

step is to assemble the chain of gears that transmit power to the Line Tracer's wheels. This stage is shown in Photo 2, the small PCB tucked away at the back of the photo being the board that houses the infra-red detectors.

Photo 3 shows the base of the Movit as

it appears a couple of stages later. Here both drive chains are complete and the rear wheel is in position. The remaining tasks are to install the two battery compartments; one for a 9V battery that powers the electronics and another to house the two 1.5V cells that power the motors. The electronics of the Line Tracer are all pre-assembled and all connections are made by way of push fit connectors so that there is no need for any soldering. This is something that will appeal to inexperienced constructors.

Photo 4 shows the completed article with its complement of batteries and ready for testing.

FOLLOW THAT LINE

The Line Tracer is designed to follow a black line drawn on a white background. The line must be 20mm wide and should not have any curves with a radius of less than 15cm. The control circuit has an adjustment for sensitivity that is tweaked to give the most reliable performance. After a few tries we managed to get our Movit to follow a line for a short distance but, as the

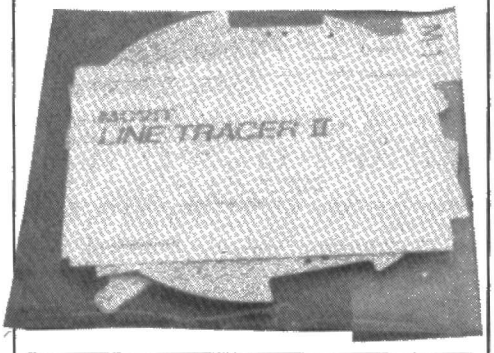


Photo 1

robot moved like the proverbial bat out of hell, it was wont to lose sight of the line quite often. This comment applies to all of the Movits that we built, in that each of them could have done with some form of motor speed control and, although we did not have time to try this, would probably work quite happily and more reliably with just a 1.5V supply to the motor.

"Movits are fun to use".

DRAWING CONCLUSIONS

Each of the Movits in the range were straightforward to put together with no requirement for any special tools. The only items required were a small cross head screwdriver and a pair of pliers. As they stand the Movits are fun to use but are not quite robots in the sense that most people would accept. However it should not be too difficult to add some additional hardware to provide a positional feedback loop and an interface to a micro. With these additions any Movit would be transformed from a toy into a serious proposition for anyone wishing to experiment with mobile robot systems.

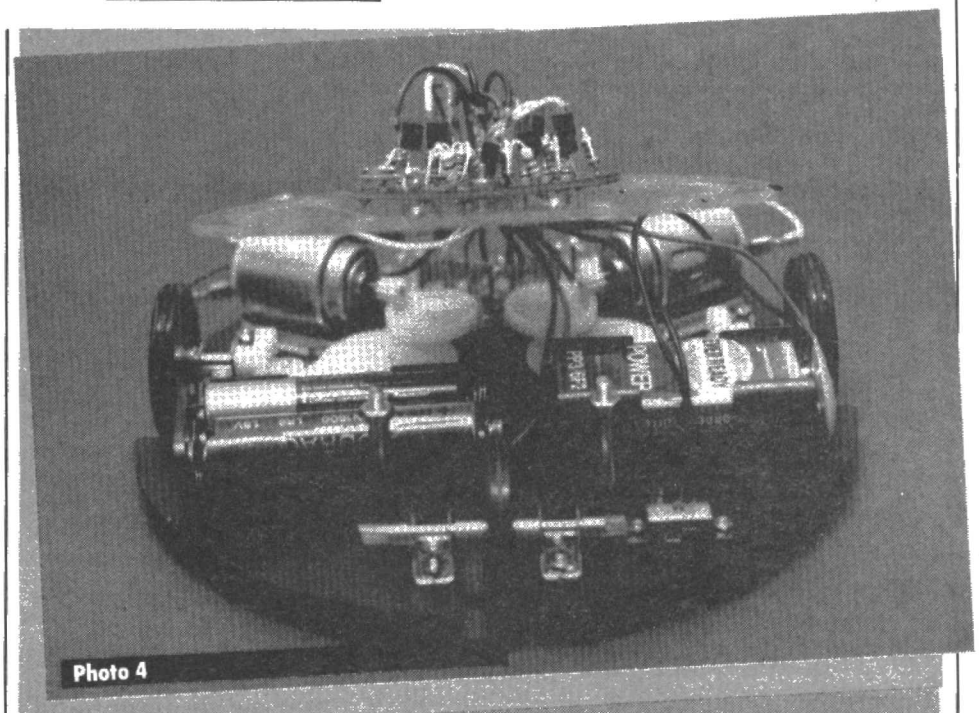


Photo 4

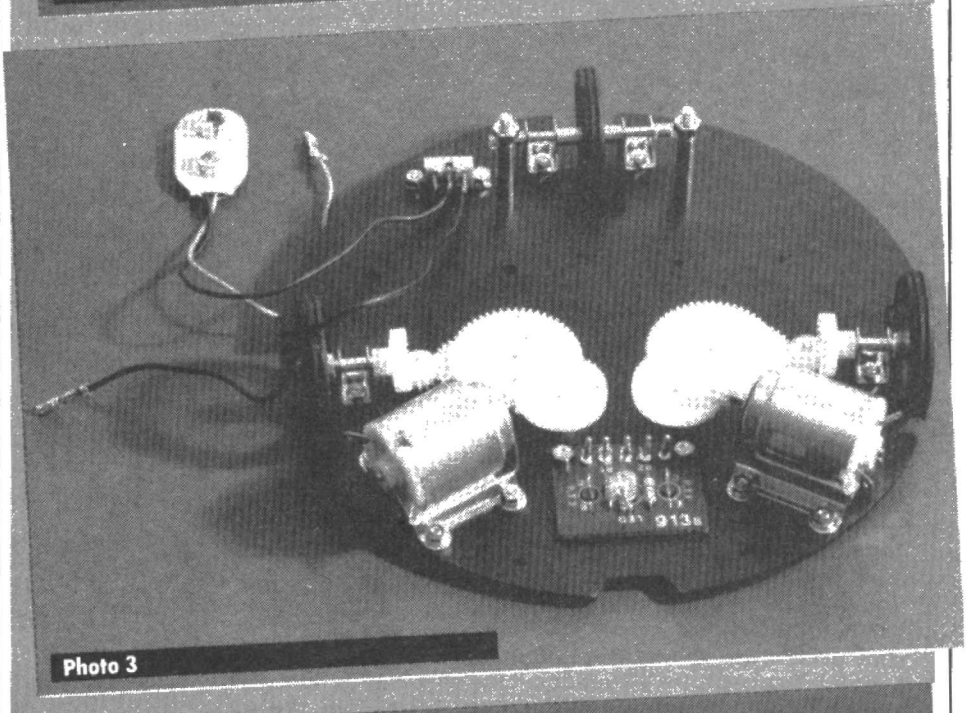


Photo 3

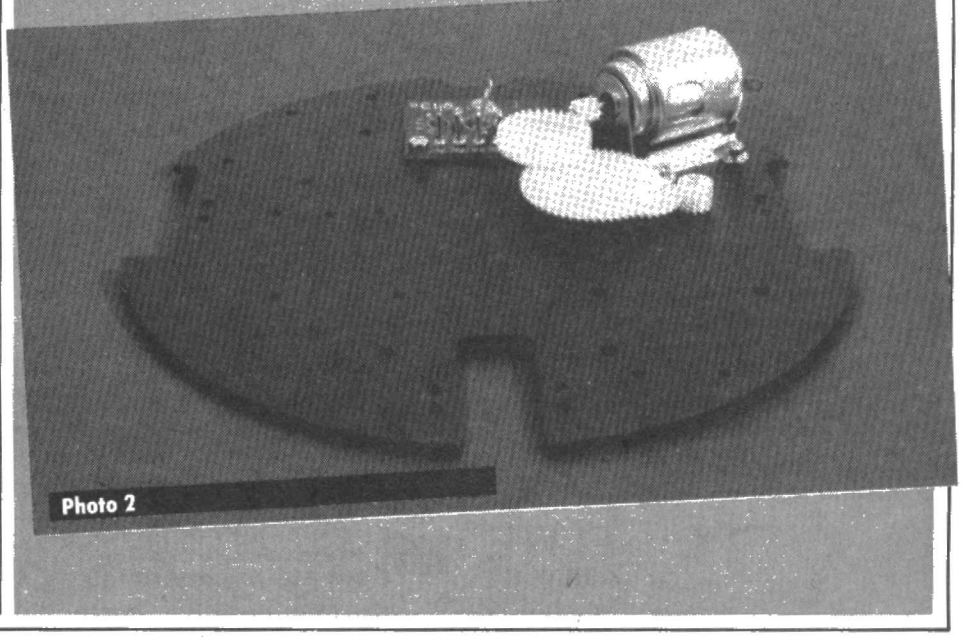


Photo 2

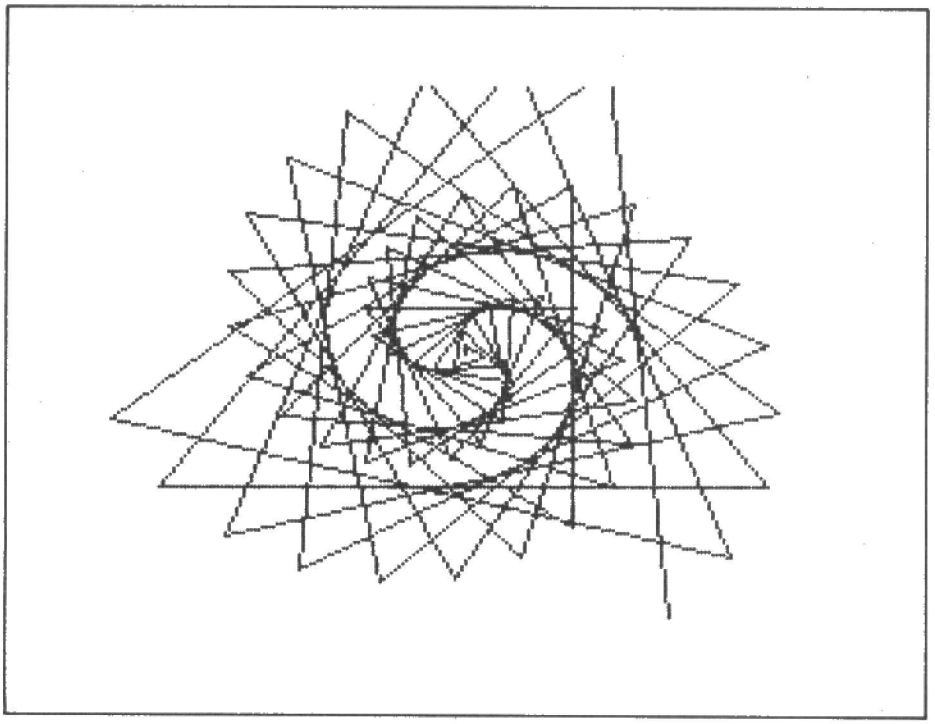
LOGO AND THE SPECTRUM

Adam Denning of Micronet 800 looks at Sinclair's implementation of LOGO. Although recognised as a powerful educational tool, LOGO has hidden power that is not readily appreciated.

Sinclair has just released one of the most important products for the Spectrum yet – the LOGO language. The package costs £39.95 and comes on cassette with two largish manuals and a quick reference card that lists all the primitives in the implementation. A primitive is the analogue of a keyword in Basic.

Unlike previous LOGOs the Sinclair version is a full LOGO with much more than the famous turtle graphics, and the power of the lesser-known parts of LOGO are readily available. In fact this cassette was written in conjunction with Seymour Pappert, the originator of LOGO.

Recognising that the Spectrum is in some ways an unusual computer, LOGO also supports its more esoteric aspects such as Microdrives and the Interface 1 RS-232 port. Of more interest to **Your Robot** readers perhaps is the ability to control an external device. Usually this is obviously a floor bound turtle, but the ingenuity of Spectrum owners is sure to cause this to be expanded and extended to other devices.



LOGO is an excellent language with which to teach young children certain concepts – not necessarily just computer-bound concepts either. In the main this is due to the LOGO turtle, which is a notional creature sitting in the middle of the screen. It is directly under the user's control and is easily manipulated with simple commands like MOVE, TURN, LEFT and RIGHT. When the turtle is asked to move, instructions prior to that will have determined whether it will draw a line along its path or not. This is done by using the analogy of a pen being held by the turtle as it moves. If the pen is down then a line is drawn; if it is up then no line is drawn. Naturally enough the primitives that control the pen are clear and very obvious: PENUP and PENDOWN.

This is the whole philosophy behind LOGO – its clarity and ease of use whilst still maintaining a laudable conciseness. Perhaps the whole turtle concept is now somewhat overblown, but when Seymour Pappert conceived the language the idea was refreshingly new. It has by no means outlived its usefulness yet, but some educationalists in particular do seem to attach more importance to it than it deserves.

This is not the case with the **whole** language – it remains a powerful addition to any programmer's collection, but the majority of people seem to be entirely ignorant of this 'hidden power'. And that's a shame.

LOGO is a procedure based language. This means that each segment of a program is written as a clearly discernable module, called a procedure. This, of course, brings us on to the argument about structured programming. LOGO is in this respect something of a special case – it is designed to teach rather more than just computing, so procedural programming is something that must be maintained. Any structured aspect is merely a sideline that happens to be in vogue at the moment.

"LOGO is an excellent language to teach young children certain concepts by virtue of its notional turtle".

A LOGO procedure is made of various primitives and user defined procedures, and is defined by typing

TO <procedure name>

This tells the LOGO interpreter that it is now learning how **to** do something, the something being a task defined by the user and referenced by the procedure name. When a procedure is being defined the cursor prompt changes (to a > in the case of a Spectrum) and text can be entered by the user. Typically a procedure may be ten lines of code, like so:

```
TO be silly
CLEARSCREEN
REPEAT 4 [SQUARE 30]
PRINT [THIS IS LOGO]
PRINT [HELLO TO ALL ELECTRONICS
AND COMPUTING READERS]
PENUP
SHOWTURTLE
MOVE 55
TURNTO 45
PENDOWN
END
```

Although not a stunningly useful definition, this example does serve to show the form of a LOGO procedure.

As anyone is prone to errors when entering or designing the text of a procedure, Spectrum LOGO incorporates a fairly advanced screen editor that is invoked with the EDIT primitive. If this word is followed by a procedure name then that procedure is listed ready for editing. Otherwise the last defined procedure is presented. The normal Spectrum cursor and delete keys can be used to alter text, and anything typed is **inserted** rather than **overtyped.**

The first thing anyone is going to do when they get hold of LOGO for the first time is play with the turtle and its graphics. This is where the first supplied manual, LOGO 1, becomes important. Apart from just describing the turtle, it also gives a general and very readable introduction to LOGO itself, explaining all about primitives, procedures and the editor. It then goes on to define some very useful introductory turtle procedures, finally bringing on the subject of variables (more correctly called Names in LOGO).

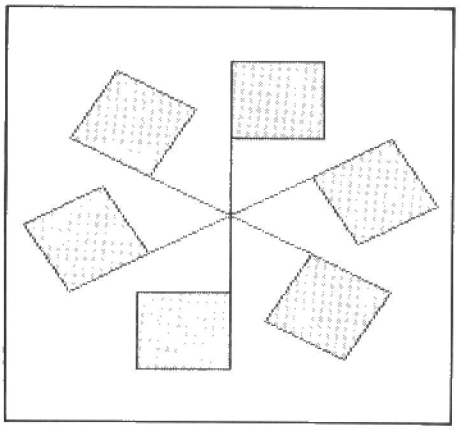
"The majority of people seem to be entirely ignorant of LOGO's hidden power".

By the end of the book these important aspects of LOGO should be understood by every reader, and some very decorative and entertaining procedures and turtle movements will have been explored too.

The second manual, LOGO 2, goes on to describe the general form of LOGO syntax and then describes every built-in primitive in detail. This is where the interest is really held, as some of the more advanced of the language's facilities are surprisingly powerful.

Unlike Sinclair's last major language release – micro-PROLOG, LOGO is fully microdrive-compatible and by using the SETDRIVE primitive with a parameter between 1 and 8 you can specify just which drive you are using. A parameter of zero sets the filing system to cassette again. Once a drive has been set all saving and loading commands will default to that microdrive cartridge, and the microdrive specific primitives such as CATALOG and ERASEFILE can be used.

There are two primitives designed for direct communication with an external turtle – STARTROBOT and STOPROBOT, but naturally these could be used to control other devices. Doubtless certain of the Z80 I/O ports are designated for particular driving purposes, and it seems likely that other turtle commands will affect these ports in some way. In fact, a closer read of the second manual reveals that once STARTROBOT has been used, LOGO uses a binary file (ie machine code) from cassette or microdrive to drive the turtle and all turtle commands **are** re-directed through this routine. The binary file must be called ROBOT. This seems an ideal area for experimentation by *E&CM* readers.



Other advanced primitives are .SETSERIAL which defines the baud rate of the RS-232 interface on Sinclair's Interface 1 and .SERIALIN and .SERIALOUT which respectively retrieve and send data via the RS-232. RECYCLE performs **garbage collection;** that is it cleans up the memory used by LOGO so that any redundant areas are re-claimed.

.BLOAD and .BSAVE load and save areas of memory as code blocks, which facilitates the use of machine code, and .RESERVE reserves space in high memory for these code blocks. .RESERVED returns the start and end addresses of a reserved block and .CALL can be used to execute machine code, taking the call address as its parameter. .DEPOSIT and .EXAMINE are identical to POKE and PEEK in Basic.

Finally, NODES gives an indication of the amount of memory free by returning the number of LOGO **nodes** available. A node occupies five bytes and is effectively a unit of memory internally to the LOGO system. .CONTENTS prints out a list of everything that LOGO **knows.** That is, the procedures and variables in memory at present. For some reason this primitive uses up a lot of notes during execution. In a similar way .PRIMITIVES outputs a list of all the LOGO primitives.

On the same subject there are numerous primitives available to re-define procedures, define procedures within other procedures, print out definitions of procedures, determine whether something is a procedure or a primitive, print out the names and values of all or selected names (variables) and erase selected or all procedures and names. There is even a primitive to remove **everything** from memory, rather like totally restarting LOGO.

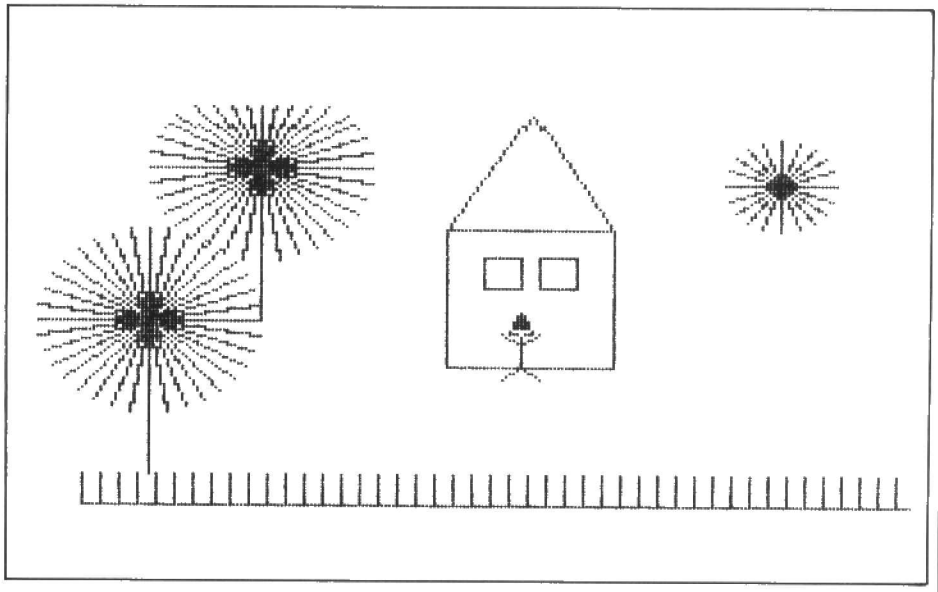
As LOGO can be used as a primarily graphic language, there are all sorts of special primitives devoted to screen handling, and a difference can be made between a purely TEXT screen and a screen with turtle. Likewise all the Spectrum attribute controls INVERSE, FLASH, OVER and BRIGHT are accessible from LOGO. The cursor can be moved to a specified position on the screen, rather like PRINT AT x,y in Basic. The very silly Spectrum beep is available from LOGO too, using the SOUND primitive.

Once the delights of the turtle have been played with, one is sure to want to experiment with the **list processing** abilities of LOGO. Initially this will involve nothing more than discovering what OUTPUT, PRINT, READCHAR, DEADLIST and SHOW can do. The first two are fairly obvious, while the next two are essentially Basic's INKEY$ and INPUT statements.

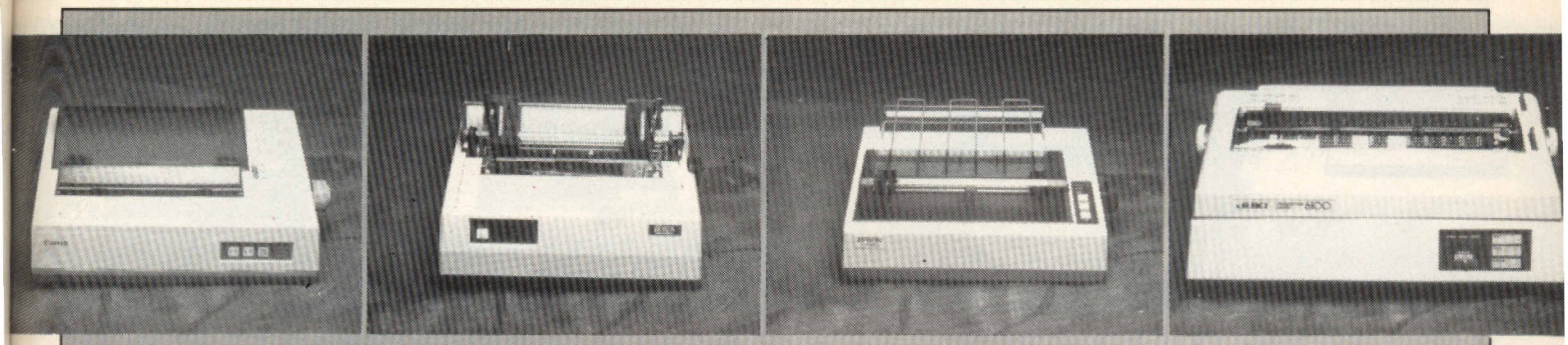
"Each segment of LOGO program is written as a clearly discernible module".

The final one has no direct analogue in Basic, but prints out anything that the user has just input.

It is very easy to take LOGO's list processing capabilities a lot further, and the many primitives dedicated to this make it a positive delight. That, in essence, is LOGO: a delight to use. A recommended purchase for anyone of absolutely any age!



The printouts on this page are examples of what can be achieved with LOGO programming.



GETTING INTO PRINT

Even a low cost printer represents a considerable capital outlay and choosing a device to meet both you and your budget and application means answering a number of important questions. Peter Luke offers a printer buyers' guide and surveys a selection of the many printers on the market.

Although printers have come down in price over the past few years, even a modest model represents a considerable capital outlay – more than many home computers in fact. This means that it is important to ensure that the right printer for the job is selected. But before even this question is addressed you should ask yourself whether or not you really want a printer at all.

Many peoples' prime use for a printer will be to produce hardcopy listings of programs during the writing and debugging of software. Now, while it is undoubtedly a help to have hardcopy output when undertaking this sort of work, are the advantages worth paying anything between £200 and £400 for? If you're only likely to want to use the printer in this way once or twice a month then the answer would appear to be no.

However, this assessment of whether or not to buy a printer leaves out one important point which is that, having purchased a printer, many other applications are bound to come to mind – not least of which is wordprocessing.

Having decided to buy, the next problem is how to select a model that is suitable for your application. One of the major factors here will be how much money you can afford to spend but another important point to bear in mind is the sort of quality that you will need from the printer. Sinclair's ZX printer proved that it was possible to provide paper output for as little as £50 but the quality of the results was hardly suitable for printing out a report for the MD! On the other hand, there is little point in parting with vast sums of money if the output will be for your eyes only – if you can tell an 'm' from an 'n' on the output you'll probably be happy.

Just to reinforce the above – think carefully about the application before selecting the type of printer you require.

Types of type

Printers can very broadly be divided into two classes – impact (when, as in a typewriter the printed character is formed by physically striking a ribbon with the print head to transfer the ink to the paper) and non-impact (here the image is usually formed by a thermal process). Each of these two broad types can be broken down into a number of sub-groups.

We'll start off at the Rolls Royce end of the market with a look at daisy wheel printers. These peripherals feature a print mechanism that consists of a wheel around the circumference of which all the available characters appear in relief. This wheel is spun until the required character is in the print position upon which it is struck by a solenoid driven hammer which prints the image onto the paper via an inked ribbon.

"The first question to ask is whether or not you need a printer at all".

This type of printer offers the highest quality of output but is rather slow in operation. Although they do tend to be in the upper price range, daisy wheel printers can be bought quite reasonably. These lower cost models are usually electric typewriters which, with the addition of an interface, can be used with a computer.

Other types of impact printer that follow this theme are golfball machines; the most notable of these being the models produced by IBM.

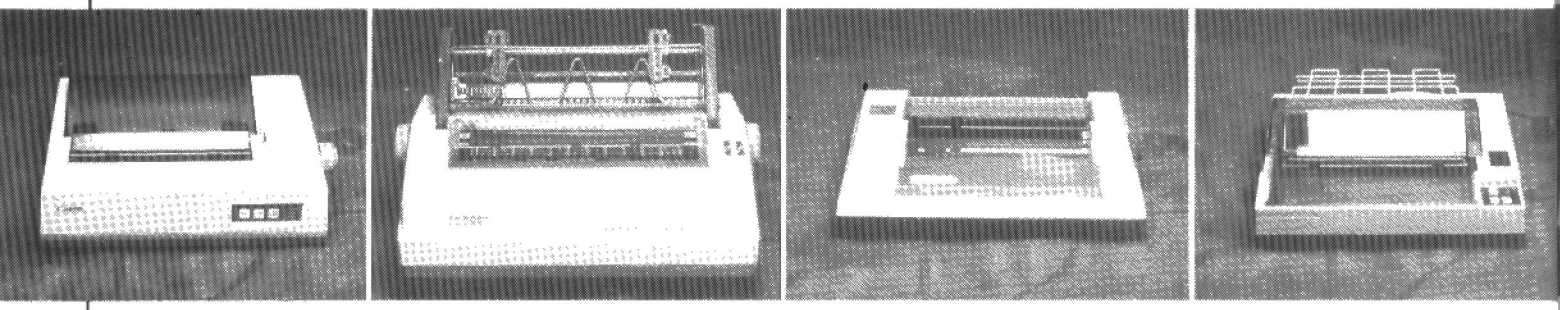
The type of printer likely to be selected by most home micro users is a dot matrix design. These can either be of an impact or thermal design. In both, the printed image is formed by a series of individual dots arranged as a matrix. The number of dots featured in the matrix will determine the quality of the finished result.

In an impact dot matrix printer the print head consists of a cluster of print needles which hammer an inked ribbon to form a character. In a thermal model the hammers are replaced by a set of wires which rapidly heat up as required the 'burn' material from a special ribbon into the paper. Results obtained from impact printers are in general superior to those obtained from a thermal design and thermal printers, if they are to work at their best, should be used with special paper. This removes the need for the print ribbon, instead the heat from the print head will chemically change the surface of the paper to produce the image.

The ZX printer referred to above used this sort of approach but the quality of print obtained from a typical thermal dot matrix printer is far superior to the Sinclair printer.

Hook up headaches

In theory there are just two types of printer interface. The Centronics standard for the parallel connection of printers and the RS232 format for serial links. In practice anyone who has tried to interface a printer to a computer will know that this is not the case. Most manufacturers of computer equipment seem to treat the respective standards as the starting point for their design rather than as rigid definition of the final implementation of the interface. This extends beyond the electrical specifications of the interface but to the types of



connectors used at either end of the link, although these too are defined in the standards.

It is therefore essential to ensure that a printer is compatible with your computer and preferable to get your dealer to prepare a lead for you – even the most dedicated electronics hobbyist would baulk at the idea of preparing a 20-way Centronics cable.

The compatibility problems when hooking up an RS232 printer are even more acute as not only do the electrical connections have to be right but also the format of the transmitted data has to be set (baud rate, number of stop bits, odd or even parity – the list goes on).

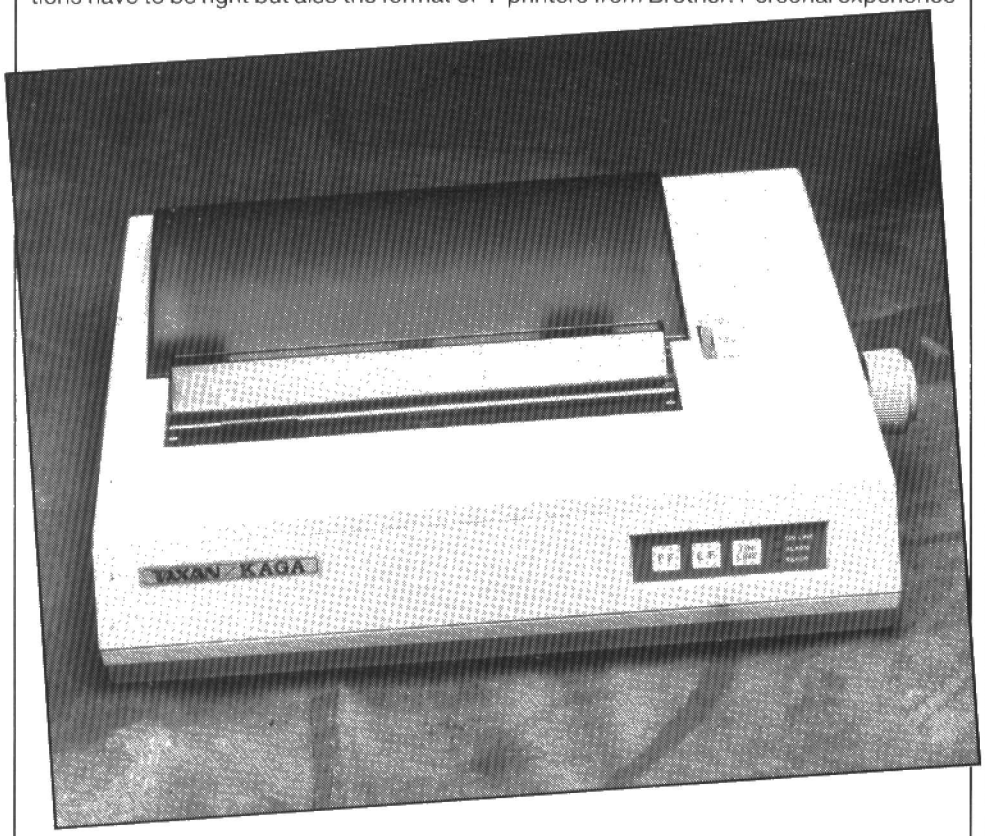
In general then, unless there is a specific requirement for a serial link, the Centronics interface is to be preferred. The parallel interface is also favoured on the grounds that most computers default to a setting for a parallel printer and selecting the serial option involves extra effort on the part of the user.

This situation was fine until recently in that RS232 interfaces were usually an optional extra on most printers and most were thus used in the parallel mode. Lately, however, the increasing emphasis on communications via modems (which require a serial data stream to drive them) has meant that terminals and printers with only an RS232 port are beginning to appear. Most notable amongst these are the low cost printers from Brother. Personal experience has shown however that interfacing these to computer systems is not as easy as the manuals would tend to suggest.

"In order to avoid compatability problems it is essential to see the printer in use both with your computer and the software you wish to run on it".

The problems of compatibility extend beyond the hardware of a system in that the software driving the printer must also be compatible with the hardware. There is usually no problem in printing straight alpha/numeric data, the difficulties come when you wish to make use of the special features of a printer (boldface, underline etc). These are invoked by outputting a non-printable control character from the computer.

The specific codes used by printers can vary and once again it is as well to ensure that your computer is compatible with any specific printer or that the software you are using allows you to tailor the command codes to the requirements of the printer.



The Kaga Taxan is available from Watford Electronics and offers excellent value for money.

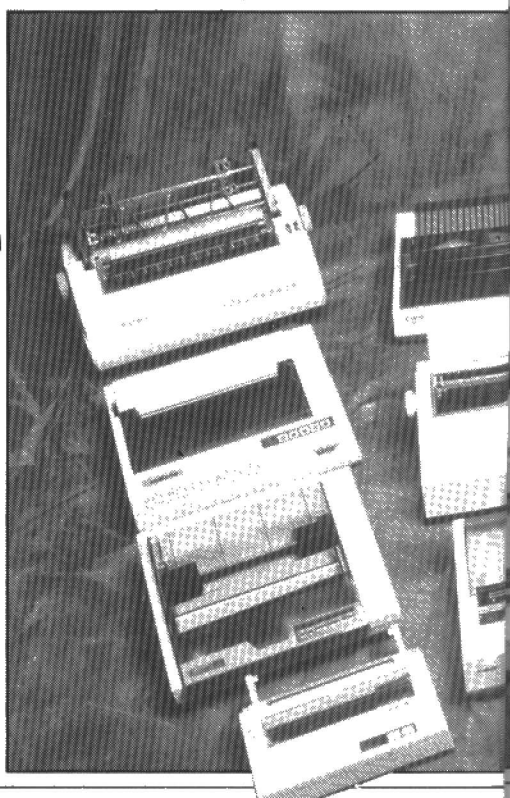
The bright side

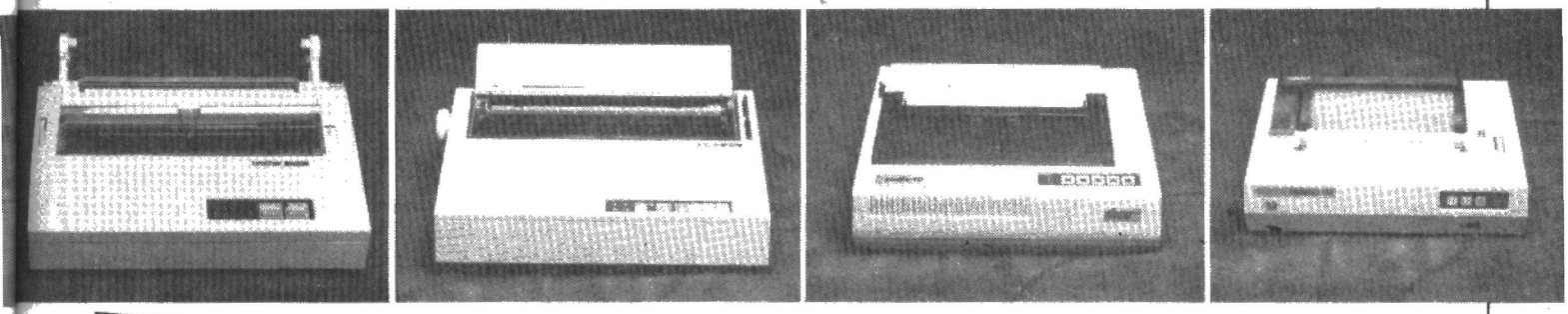
The above was not designed to put you off buying a printer but merely to highlight some of the problems that can be encountered. As always the best advice is to ask your dealer to demonstrate the printer of your choice in conjunction with the type of computer that you own and running the software that you are going to be using on the system.

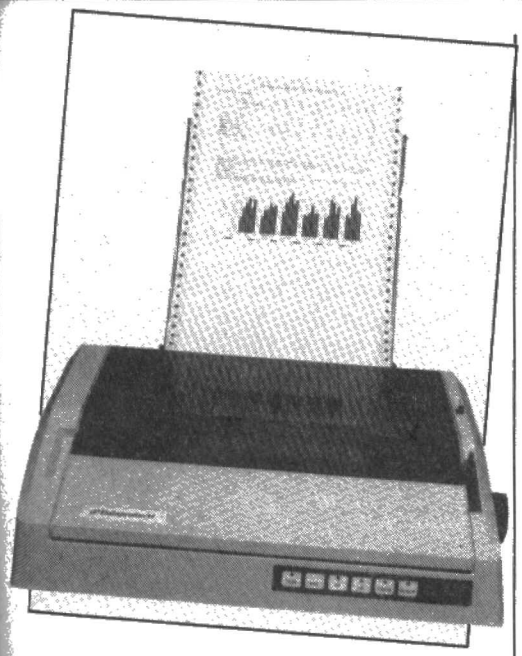
Naming names

Many printer manufacturers produce a range of printers covering models designed for the home market at the low end to models designed with the business user firmly in mind. We do not have the

Below. This collection of printers shows the vast rang







room to mention all the designs produced by the various companies nor indeed to mention all those firms with a toe in the printer market. What follows is a guide to some of the printers available in the £100 to £400 price range.

Brother have been making quite a name for themselves over the past year or so. Their EP22 typewriter/terminal broke new grounds in terms of quality, low cost print and, if this was not enough, followed this with the launch of the EP44. This offered much higher definition but still managed to come onto the market at just over £200. Both these models offered a keyboard as well as a printer but the recently launched HR-5 offers the option of buying just a printer. The printer retails at a little over £150 and offers excellent value for money. It can be powered from batteries or from an optional mains power unit (make sure that you remember to switch the printer off after use as it does not feature an auto power off facility and draws power even when dormant as we found to our cost).

The HR-5 is a thermal printer and can be used with specially coated paper or with ordinary stock in conjunction with an ink ribbon. It supports bi-directional printing and can cope with 80 col. output.

It features an RS232 interface as standard which should make it an ideal choice for owners of the BBC micro and Dragon to name just two popular computers that support an RS232 port. We tested the HR-5 in conjunction to print out BASIC programs quite happily although in the time allowed could not sort out some problems relating to its use with the OS9 OS.

At the other end of the scale Brother produce the HR-15 daisy wheel printer which retails at around £340. This is excellent value for a daisy wheel and if your application demands quality rather than speed the HR-15 is certainly worth a look. Brother also produce a number of other printers and a look at their catalogue should prove a useful exercise if you are in the market for a printer.

At the lower end of the price scale Alphacom have two printers that are worth trying out. These are the '81 and '41 which support 80 and 40 column printing respectively. Although not capable of letter quality output they should prove adequate in many home applications and have the advantage of not running up too much of an overdraft.

"Some real bargains are available at present".

The name Epson is almost synonomous with printers and there are more than a few Epson clones around on the market. The company has two printers that are of interest to home users – the RX and FX 80. The RX 80 retails at around £300, prints at 100 CPS and offers a very credible performance. The FX 80 stretches our budget limit, and indeed that of some purchasers as it costs around £450. It offers a host of features including a fast 160 CPS print speed. It also offers a variety of special features including NLQ (near letter quality) output, although in this mode print speed is halved at 80 CPS.

Other notable names in the printer industry are Mannesman, Juki and Qume. Each of these produce printers that neatly straddle our declared price range. Canon too have recently entered the printer market and amongst other models offer the PW-1080A. This just scrapes in under our £400 limit and offers a host of special facilities in addition to zooming along at 160 CPS in standard mode.

A name that many people will not have heard of before is Kaga. Their Taxan KP810 printer has the attraction of offering Epson compatible control codes and has a number of different print modes to suit varying applications. Standard speed is 160 CPS although an 80 CPS NLQ mode can be selected at switch on. Normal print width is 80 cols, although this can be stretched to 136 for those with vision in excess of 20/20.

models available.

TERMI 2
BBC Model B ROM £33.35
COMMUNICATOR
BBC Model B ROM £69.00
Both from Computer Concepts

These two ROMs from Computer Concepts turn the BBC micro into a communications terminal. TERMI 2 is a new version of the familiar TERMI ROM, which emulates a DEC VT52 terminal, and Communicator emulatesd either a VT53 or DEC VT100 terminal. The higher price of Communicator reflects its additional features.

TERMI 2 has only four commands (listed by HELP): GO, LOAD <fsp>, SAVE <fsp>, and SETUP. LOAD and SAVE are used to load and save text or data to the terminal; the material is then transmitted using the GO command, under conditions which have previously been SETUP. SETUP allows the user to choose state, printer commands, baud rates for send and receive (75 to 19200 – compared to 75 to 4800 on TERMI 1), word size, character set, protocol, handshaking, screen mode, etc. etc.

The TERMI 2 terminal, the VT52, is described as 'semi-intelligent', with limited menu facilities; it is however very easy to use with a simple cursor operation to delimit the parameters. The Communicator ROM goes much further. Communicator is a full 16K ROM with extensive menus, function key operation, and a number of new OSBYTE calls. Keywords include CONFIGURE, GO, HELP, INITIALISE, KEYS, LOAD, RESET and SAVE. There are three modes of operation: COMMAND, SETUP, and COMMUNICATIONS. COMMAND mode is the highest level of operation and provides the main interface to operating systems commands, as well as access to a small set of keywords to drive the terminal software. SETUP mode is plit up into three pages, to set parameters, tabs, and function key strings. SETUP is similar in operation to that in TERMI 2, with similar baud rates available.

Communicator can be used with the Wordwise and VIEW wordprocessors, and is ideal for anyone who needs to communicate text or data quickly and cheaply over standard electronic mail systems such as Telecom Gold, or with certain bulletin boards; it should be remembered however that the software is not suitable for Prestel or Micronet.

SOFTWARE FILE

A BUYER'S GUIDE TO UTILITY SOFTWARE

MACHINE	PRODUCT	PRICE	FORMAT	MEMORY	SUPPLIER	COMMENT
COMMUNICATIONS						
APPLE II/IIE	COMMS PACK	49.95	D		PRISM	PRESTEL/MICRONET SOFTWARE – SPECIAL COMMS CARD VERSION
APPLE II/IIE	COMMS PACK	29.95	D		PRISM	PRESTEL/MICRONET COMS SOFTWARE
BBC	COMMUNICATOR	69.00	R	16K	COMPUTER CONCEPTS	DEC VT100 EMULATOR
BBC	TERMI	33.35	R	8K	COMPUTER CONCEPTS	DEC VT52 EMULATOR
BBC	TERMI 2	33.35	R	8K	COMPUTER CONCEPTS	ADVANCED VERSION OF TERMI
BBC	COMMSTAR	34.00	R	8K	PACE	WILL ACCESS PRESTEL, UNLIKE ABOVE
BBC	COMMS PACK	19.95	R		PRISM	PRESTEL/MICRONET COMMS SOFTWARE
BBC	COMMS PACK	14.95	C		PRISM	AS ABOVE
CBM64	COMMUNICATION CARTRIDGE	43.00	CR		MICRONET	FOR ANY STANDARD V21 OR V23 MODEM – USABLE ON MICRONET/PSS/TELECOM GOLD
RML380Z	COMMS PACK	34.95	D		PRISM	PRESTEL/MICRONET SOFTWARE
SPECTRUM	PRACTICAL WIRELESS RADIO PROGRAMS 3	3.75	C	16K	PRACTICAL WIRELESS	7 PROGRAMS INC. DISTANCE AND BEARINGS, TV TEST PATTS, C/SIGN FILING SYS – FOR HAMSTERS
SPECTRUM	PW RADIO PROGRAMS 5	3.75	C	16K	PRACTICAL WIRELESS	6 MORE PROGRAMS FOR HAMSTERS
TANDY 1	COMMS PACK	24.95	C		PRISM	PRESTEL/MICRONET SOFTWARE
TANDY 3	COMMS PACK	29.95	C		PRISM	PRESTEL/MICRONET SOFTWARE
ZX81	PW RADIO PROGRAMS 1	3.50	C	1/16K	PRACTICAL WIRELESS	FOUR 1K AND FOUR 16K PROGS FOR RADIO HAMS
ZX81	PW RADIO PROGRAMS 2	5.75	C	16K	PRACTICAL WIRELESS	STRUCTURED MORSE LEARNING COURSE
ZX81	PW RADIO PROGRAMS 4	3.75	C	16K	PRACTICAL WIRELESS	6 PROGRAMS INC. ANTENNA FEED CALCULATIONS

EXCITING ADDITIONS FOR YOUR HOME COMPUTER

KEYBOARD with ELECTRONICS for ZX SPECTRUM

★ Full size, full travel keyboard that simply plugs into expansion port on your Spectrum. ★ Offers single key selection of all major multi-key functions. ★ Extends port for other peripherals. ★ Can accept Atari-type joysticks (optional extra — order 2 of FG66W, £1.36 each and note that case will require cutting).
Three kits needed to build unit: Order LK29G, LK30H & XG35Q. Total price £39.95. Full construction details in Project Book 9 XA09K 70p. Also available ready-built. Order As XG36P. Price £44.95.

KEYBOARD with ELECTRONICS for ZX81

★ Full size, full travel keyboard that's easy to add to your ZX81. ★ No soldering in ZX81; simple instructions make it easy to fit. ★ Makes Shift Lock, Function & Graphics 2 single key selections.
Complete kit (excl. case) LW72P Price £23.95. Case XG17T £4.95. Full construction details in Project Book 3 XA03D. Price 70p. Ready-built in case XG22Y. Price £32.50.

MODEM

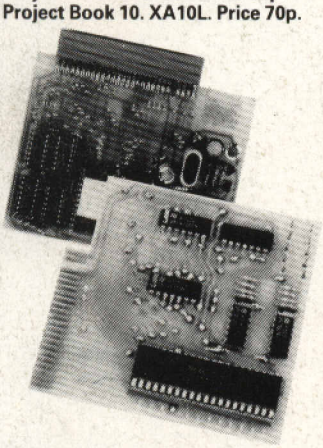
A CCITT standard modem that connects directly to your telephone line via a BT approved transformer. Transmits and receives simultaneously on European standard frequencies at 300 baud. May be used to talk to any other 300 baud European standard modem including the Maplin Computer Shopping modem on 0702 552941 and any British Telecom Datel 200/300 Service modem. The modem's computer interface is RS232 compatible.
Complete kit (excl. case) LW99H. Price £44.95. Case YK62S £9.95. Full construction details in Project Book 5 XA05F Price 70p.

INTERFACES for MODEM

Interfaces are now available for the following machines: Commodore 64, Dragon 32, Oric, Spectrum, VIC20 and ZX81. Each is complete with a Machine Code Communications program. The BBC micro needs no interface and a suitable program is on Maplin catalogue page 15 or Project Book 8, page 59.

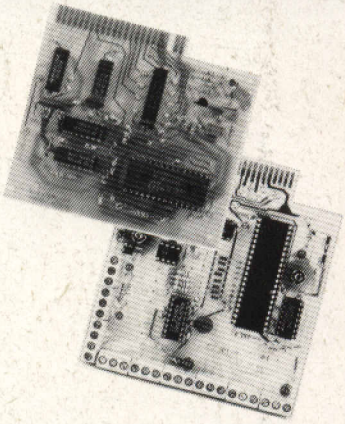
Computer	Order	Details	Price
64/VIC20	LK11M	Book 7	£9.45
Dragon 32	LK12N	Book 8	£18.95
Oric 1	LK40T	Book 10	£13.95
Spectrum	LK21X	Book 8	£19.95
ZX81	LK08J	Book 7	£29.95

Project Book 7 XA07H. Price 70p. Project Book 8 XA08J. Price 70p. Project Book 10. XA10L. Price 70p.



ZX81 I/O PORT

★ Provides two bi-directional ports for 16 input or output lines. ★ One buffered output which can interface directly to CMOS. ★ On board address selection permits expansion to six ports with two boards.
Complete kit LW76H. Price £10.49. Full construction details in Project Book 4 XA04E. Price 70p.



MAPLIN TALK-BACK SPEECH SYNTHESISERS

★ Unlimited vocabulary with allophone (extended phoneme) system. ★ Can be used with unexpanded Oric 1, VIC20 or ZX81 as it does not require large areas of memory. ★ Speech may be easily added to programs. ★In VIC20 version speech output is direct to TV speaker with no additional amplification needed.

Computer	Order	Details	Price
Oric 1	LK28F	Book 9	£23.95
VIC20	LK00A	Book 6	£22.95
ZX81	LK01B	Book 6	£19.95

Project Book 6 XA06G. Price 70p. Project Book 9 XA09K. Price 70p.

DRAGON 32 I/O PORT

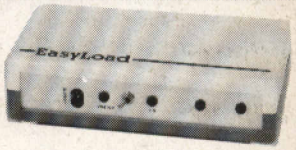
★ Provides two TTL & 3-state bus compatible 8-bit ports, ★ Four norm/inv. latched ports, ★ Two relay switched ports ★ And two opto switched ports. ★ Module plugs directly into cartridge socket and is fully programmable from BASIC.
Complete kit LK18U. Price £14.95. Full construction details in Project Book 8 XA08J. Price 70p.

MAPLIN CATALOGUE

Full details of all Maplin's projects and electronic components in our huge 480 page catalogue. On sale now in all branches of W.H. Smith price £1.35. Or send £1.65 (incl. p&p) to our mail order address.

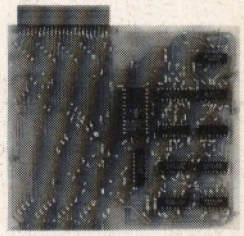
OTHER PROJECTS

For full details of our other computer-related projects please see the relevant project book as below:
BBC Motherboard — Book 11. ZX81 Sound on TV — Book 6. ZX81 Extendi-RAM — Book 9. VIC20 Extendiboard — Book 9. Dragon 32 Extendiport — Book 10. TTL/RS232 Interface — Book 9. Project Book 6 XA06G. Price 70p. Project Book 9 XA09K. Price 70p. Project Book 10 XA10L. Price 70p. Project Book 11 XA11M. Price 70p.



SPECTRUM EASYLOAD

★ Greatly reduces cassette LOADing & SAVEing problems on Spectrum. ★ Battery powered, no bus connections. ★ Charging from Spectrum PSU. ★ SAVE & LOAD indicators.
Complete kit (excl case) LK39N. Price £9.95. Full construction details in Project Book 10 XA10L. Price 70p.

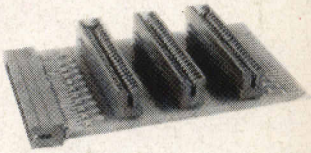


ZX81 HI-RES GRAPHICS

★ Full 256 x 192 fine pixel display with normal or inverted video. ★ Draws lines, circles and triangles, fills and textures. ★ Up to 32 user defined graphics. ★ Operates directly from extended BASIC.
Complete kit LK23A. Price £27.50. Full construction details in Project Book 9 XA09K. Price 70p.

ZX81 SOUNDS GENERATOR

★ Turns your ZX81 into a mini-synthesiser. ★ 3 programmable tone generators. ★ 3 programmable attenuators. ★ Noise generator with 3 pitch levels for special effect sounds. ★ Single address access with PEEK & POKE. ★ Connects directly to extension board or expansion port socket with extra socket (order RK35Q £2.20) ★ Requires separate amp and speaker.
Complete kit LW96E. Price £13.49. Full construction details in Project Book 5 XA05F. Price 70p.



ZX81 EXTENSION BOARD

★Plugs directly into ZX81 expansion port. ★ Accepts a 16K RAM pack and three other plug-in modules simultaneously. Parts are sold separately as follows:
PCB GB08J. Price £2.40. Edge Connectors (4 needed) RK35Q. Price £2.20 each. Track pins (1 pack needed) FL82D. Price 85p per pack of 50.

MAPLIN
ELECTRONIC SUPPLIES LTD

Maplin Electronic Supplies Ltd. Mail Order: P.O. Box 3, Rayleigh, Essex SS6 8LR.
Tel: Southend (0702) 552911. • Shops at: 159-161 King Street, Hammersmith, London W6. Tel: 01-748- 0926. • 8 Oxford Road, Manchester. Tel: 061-236-0281. • Lynton Square, Perry Bar, Birmingham. Tel: 021-356-7292. • 282-284 London Road, Westcliff-on-Sea, Essex. Tel: 0702 554000. • 46-48 Bevois Valley Road, Southampton. Tel:0703 25831. All shops closed all day Monday.
All prices include VAT and carriage. Please add 50p handling charge to orders under £5 total (except catalogue).